VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Algorithms for flow problems with stick-slip boundary conditions in three space dimensions

Algoritmy pro úlohy proudění se skluzovou podmínkou

ve třech prostorových dimenzích

Bc. Vladimír Arzt

Diploma Thesis

Supervisor: prof. RNDr. Radek Kučera, Ph.D.

Ostrava, 2021

Abstrakt

Diplomová práce se zabývá Navier-Stokesovou úlohou řešenou pomocí metody konečných prvků ve dvou a třech prostorových dimenzích. Obsahuje formulaci úlohy a její slabou formulaci se skluzovou podmínkou. Výsledná úloha obsahuje dvě nelinearity. První způsobená konvektivním členem je linearizována za pomoci Ossenových iterací a druhá způsobená přítomností nelineární skluzové podmínky je řešena semihladkou Newtonovou metodou. Je provedena kontrola konvergence Ossenových iterací ve dvou a třech dimenzích. Následují experimenty s různými typy předpodmínění BiCGstab řešiče, který je použit k řešení vnitřních úloh uvnitř Ossenových a také Newtonových iterací. Testy byly provedeny na různých oblastech s různými okrajovými podmínkami. V dodatku je popsáno odvození vektorizovaných algoritmů pro sestavení matic tuhosti a vektoru pravé strany pro Navier-Stokesovy úlohy, kde se mimo lineárních bázových funkcí používá i funkcí bublinkových.

Klíčová slova: metoda konečných prvků, Navier-Stokesova úloha, semihladká Newtonova metoda, skluzová okrajová podmínka

Abstract

The diploma thesis deals with the Navier-Stokes problem solved using the finite element method in two and three spatial dimensions. It contains the formulation of the problem and its weak formulation with a stick-slip boundary condition. The resulting problem contains two nonlinearities. The first caused by the convective term is linearized by Ossen iterations and the second caused by the nonlinear stick-slip condition is solved by the semi-smooth Newton method. The convergence of the Ossen iterations in two and three space dimensions is checked. The following are experiments with different the preconditioners of the BiCGstab solver, which is used to solve internal problems within Ossen's and also Newton's iterations. The test were performed on different domains with different boundary conditions. The appendix describes the derivation of vectorized algorithms for the construction of stiffness matrices and the right-hand vector of the Navier-stokes problem, where in addition to the basic linear basis functions, the bubble function is also used.

Key Words: finite element method, Navier-Stokes problem, semi-smooth Newton method, stick-slip boundary condition

Acknowledgement

I would like to thank my thesis supervisor prof. RNDr. Radek Kučera, Ph.D for his patience, willingness, comments and advice, which directed this work in a better direction. I would also like to thank my family, who did not have it easy during my studies, and also my classmates, who offered a certain relaxation during the common sleepless nights spent writing our final theses.

Contents

List of symbols and abbreviations

Ω	–	domain Ω
$\partial\Omega$	–	domain Ω boundaries
$\bar{\Omega}$	–	domain Ω seal
$C(\bar{\Omega})$	–	space of functions that are continuously extendable up to the boundary $\partial\Omega$
$C_0^\infty(\Omega)$	–	space of infinitely differentiable functions in the Ω, whose trace is zero at the boundary $\partial\Omega$
$L^2(\bar{\Omega})$	–	space of Lebesque integrable functions in the square
$H^1(\Omega)$	–	Sobolev space of functions
$H_0^1(\Omega)$	–	Sobolev space of functions, whose trace is zero at the boundary $\partial\Omega$
$P_1(T)$	–	polynomials of at most the first degree on the element T
∇	–	gradient
$\nabla\cdot$	–	divergence ($\nabla\cdot = \text{div}$)
Δ	–	Laplace operator ($\Delta = \nabla^2 = \nabla \cdot \nabla = \text{div grad}$)
$\|\| \cdot \|\|$	–	Euclidean norm
FEM	–	finite element method

List of Figures

List of Tables

Listings

1 Introduction

The beginning of numerical modeling of fluid flow dates back to the 1930s. It has undergone great development since the 1950s with the rise of computers, especially in the twenty-first century. A large part of numerical models deals with the solution of the Navier-Stokes equations which are named after French engineer and physicist Claude-Louis Navier and Anglo-Irish physicist and mathematician George Gabriel Stokes.

The Navier-Stokes equations are a set of partial differential equations which describe the motion of viscous fluid substances. Unlike Euler's equations, the Navier-Stokes equations take viscosity into account while the Euler equations model only inviscid flow. Note that Navier-Stokes equations can be solved only by numerical methods. The Navier-Stokes are the equations of conservation of mass, momentum and energy for Newtonian fluids as follows:

$$\partial_t \rho + \nabla \cdot [\rho \boldsymbol{u}] = 0, \tag{1.1}$$

$$\partial_t(\rho \boldsymbol{u}) + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot [\eta \nabla \boldsymbol{u} + \frac{1}{3}\eta(\nabla \cdot \boldsymbol{u})\mathbf{I}] + \nabla p = \rho \boldsymbol{f}, \tag{1.2}$$

$$\partial_t(c_p \rho T) + c_p \rho \boldsymbol{u} \cdot \nabla T - \nabla \cdot [\lambda \nabla T] = \boldsymbol{h}, \tag{1.3}$$

where $\boldsymbol{u}$ is the velocity, ρ the density, p the pressure, and T is the temperature of the fluid. The fluid is also characterized by dynamic viscosity $\eta > 0$, heat capacity $c_p > 0$, and heat conductivity λ. The term $\boldsymbol{f}$ characterizes the volume force and $\boldsymbol{h}$ the heat source. We will consider the isothermal flow, so we can neglect the third equation (1.3). We also consider the incompresible fluid, then the density will by homogeneous such that $\rho_0 := \rho = \text{const.}$, so that (1.1) reduces to the incompressibility constraint

$$\nabla \cdot \boldsymbol{u} = 0. \tag{1.4}$$

With (1.4) For simplicity we set $\rho_0 = 1$, and with (1.4) we can (1.2) write as follows:

$$\alpha \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot [\nu \nabla \boldsymbol{u}] + \nabla p = \boldsymbol{f}, \tag{1.5}$$

where $\nu = \frac{\eta}{\rho_0}$ is the kinematic viscosity and $\alpha \geq 0$ is an arbitrary konstant and for our case will be $\alpha = 0$. Then we can write our form of Navier-Stokes equations, which we will solve in this thesis as follows:

$$-\nu \Delta \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \boldsymbol{f}, \tag{1.6}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{1.7}$$

Navier-Stokes equations (1.6)-(1.7) can be solved by various numerical methods, each with its pros and cons. The most used are the finite difference method, which is easy to implemetation, but has problem with curved boundaries, mesh adaptation, and with stability and convergence

analysis, the finite volume method, which is used by CFD programs like Ansys or OpenFoam, but it has problem with unsctructed meshes and difficult stability and convergence analysis, and finally the finite element method, which we will use in this thesis.

We will solve problems with Dirichlet, Neumann and stick-slip boundary condition. The Dirichlet condition is applied to those parts the boundary of the domain, that are either a solid obstacle, a wall, or an inlet. The Neumann condition wil be an "do nothing" outflow boundary condition, which ensures zero pressure at the outlet. Finally, the stick-slip condition, which, unlike the zero Dirichlet condition, allows the fluid to "tear" and slide on the surface, which is better suited physical reality.

To improve stability of FEM, we will use the P1-bubble/P1 finite element pair introduced by Arnold, Brezzi, and Fortin [6].

Next, we derive algorithms for solving problem (1.6)-(1.7) with the above boundary conditions and then perform several numerical experiments.

2 Continuous formulations of the problem

2.1 Classical formulation of the problem

Let us consider the bounded domain $\Omega \subset \mathbb{R}^d,\ d = 2, 3$, with a sufficiently smooth boundary $\partial\Omega$, that is split into three disjoint parts γ_D, γ_N, and γ_S with a non-empty interior such, that $\partial\Omega = \overline{\gamma}_D \cup \overline{\gamma}_N \cup \overline{\gamma}_S$. On this domain we model the flow of a viscous incompressible Newtonian fluid by stationary Navier-Stokes equations with the Dirichlet and the Neumann boundary conditions on γ_D and γ_N, respectively, and with the impermeability and the stick-slip boundary condition on γ_S as follows:

$$-\nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad \text{in } \Omega, \tag{2.1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{2.2}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \gamma_D, \tag{2.3}$$

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_N \qquad \text{on } \gamma_N, \tag{2.4}$$

$$u_n = 0 \qquad \text{on } \gamma_S, \tag{2.5}$$

$$||\boldsymbol{\sigma}_t + \kappa\boldsymbol{u}_t|| \le g \qquad \text{on } \gamma_S, \tag{2.6}$$

$$\boldsymbol{\sigma}_t\cdot\boldsymbol{u}_t + g||\boldsymbol{u}_t|| + \kappa\boldsymbol{u}_t\cdot\boldsymbol{u}_t = 0 \qquad \text{on } \gamma_S, \tag{2.7}$$

where $\boldsymbol{\sigma} : \partial\Omega \to \mathbb{R}^d$ is a stress defined as:

$$\boldsymbol{\sigma} = \nu\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}} - p\boldsymbol{n} \quad \text{on } \partial\Omega. \tag{2.8}$$

We are searching for a vector function $\boldsymbol{u} : \overline{\Omega} \to \mathbb{R}^d$ representing the velocity field and a scalar function $p : \overline{\Omega} \to \mathbb{R}$ representing the pressure field. The meaning of the data describing our problem is as follows: $\nu > 0$ is the dynamic viscosity, $\boldsymbol{f} : \overline{\Omega} \to \mathbb{R}^d$ are forces acting on the fluid, $\boldsymbol{u}_D : \overline{\gamma}_D \to \mathbb{R}^d$ is the fluid velocity prescribed on γ_D, i.e. the Dirichlet data, $\sigma_N : \gamma_N \to \mathbb{R}^d$ is the stress prescribed on γ_N, i.e. the Neuman data, and $g : \gamma_S \to \mathbb{R}$, $g \ge 0$, $\kappa : \gamma_S \to \mathbb{R}$, $\kappa \ge 0$ are the slip bound, and the adhesive function prescribed on γ_S, respectively. We use also this notation: $\boldsymbol{n} = \boldsymbol{n}(\boldsymbol{x}) \in \mathbb{R}^d,\ \boldsymbol{x} \in \partial\Omega$, is the unit vector of the outward normal to $\partial\Omega$ at $\boldsymbol{x}$; for $d = 2$ it is $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{x}) \in \mathbb{R}^2$ the unit tangential vector to $\partial\Omega$ at $\boldsymbol{x}$; for $d = 3$ there are $\boldsymbol{t}_1 = \boldsymbol{t}_1(\boldsymbol{x}) \in \mathbb{R}^3$ and $\boldsymbol{t}_2 = \boldsymbol{t}_2(\boldsymbol{x}) \in \mathbb{R}^3$, two unit tangential vectors to $\partial\Omega$ at $\boldsymbol{x}$ chosen such, that the triplet $\{\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2\}$ forms at $\boldsymbol{x}$ an orthonormal basis. Finally, we will define normal and tangential components of the velocity $\boldsymbol{u}$ and of the stress $\boldsymbol{\sigma}$ along boundary $\partial\Omega$. We distiugnish two situations. For $d = 2$, we introduce the size of the normal and tangential components of the velocity $\boldsymbol{u}$ by

$$u_n = \boldsymbol{u}\cdot\boldsymbol{n}, \quad u_t = \boldsymbol{u}\cdot\boldsymbol{t}$$

and the respective normal and tangential components are vectors:

$$\boldsymbol{u}_n = u_n \boldsymbol{n}, \quad \boldsymbol{u}_t = \boldsymbol{u} - u_n \boldsymbol{n} = u_t \boldsymbol{t}.$$

Analogously we defined σ_n, σ_t, $\boldsymbol{\sigma}_n$, and $\boldsymbol{\sigma}_t$. For $d = 3$, we have

$$u_n = \boldsymbol{u} \cdot \boldsymbol{n}, \quad u_{t_1} = \boldsymbol{u} \cdot \boldsymbol{t}_1, \quad u_{t_2} = \boldsymbol{u} \cdot \boldsymbol{t}_2,$$

and

$$\boldsymbol{u}_n = u_n \boldsymbol{n}, \quad \boldsymbol{u}_t = \boldsymbol{u} - u_n \boldsymbol{n} = u_{t_1} \boldsymbol{t}_1 + u_{t_2} \boldsymbol{t}_2.$$

We use also a two-component vector lying in the tangent surface defined by $\widetilde{\boldsymbol{u}}_t = (u_{t_1}, u_{t_2})$. Since

$$\boldsymbol{u}_t = \boldsymbol{0} \Leftrightarrow \widetilde{\boldsymbol{u}}_t = \boldsymbol{0}, \quad ||\boldsymbol{u}_t|| = ||\widetilde{\boldsymbol{u}}_t||, \quad \boldsymbol{u}_t \cdot \boldsymbol{u}_t = \widetilde{\boldsymbol{u}}_t \cdot \widetilde{\boldsymbol{u}}_t, \tag{2.9}$$

we denote $\widetilde{\boldsymbol{u}}_t$ usually by the symbol $\boldsymbol{u}_t$. Similar definitions and notations can be introduced also for $\boldsymbol{\sigma}$. Note that the norm on left side of equation (2.9) is the Euclidean norm in $\mathbb{R}^3$ while the norm on the right side is the Euclidean norm in $\mathbb{R}^2$. Similar observations for $d = 2$ allow us to replace the Euclidean norm in $\mathbb{R}^2$ by the absolute value and instead of $\boldsymbol{u}_t \in \mathbb{R}^2$ to use $u_t \in \mathbb{R}$.

We will now interpret the stick-slip condition (2.6), (2.7) for $d = 2$.

Lemma 1 *Let $d = 2$, $\boldsymbol{x} \in \gamma_S$. The condition*

$$|\sigma_t(\boldsymbol{x}) + \kappa(\boldsymbol{x}) u_t(\boldsymbol{x})| \leq g(\boldsymbol{x}), \tag{2.10}$$

$$\sigma_t(\boldsymbol{x}) u_t(\boldsymbol{x}) + g(\boldsymbol{x}) |u_t(\boldsymbol{x})| + \kappa(\boldsymbol{x}) u_t^2(\boldsymbol{x}) = 0 \tag{2.11}$$

is satisfied iff

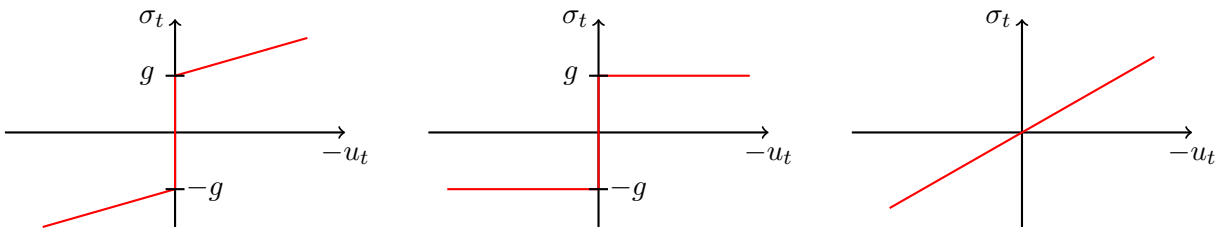
$$u_t(\boldsymbol{x}) = 0 \Rightarrow |\sigma_t(\boldsymbol{x})| \leq g(\boldsymbol{x}), \tag{2.12}$$

$$u_t(\boldsymbol{x}) > 0 \Rightarrow \sigma_t(\boldsymbol{x}) = -g(\boldsymbol{x}) - \kappa(\boldsymbol{x}) u_t(\boldsymbol{x}), \tag{2.13}$$

$$u_t(\boldsymbol{x}) < 0 \Rightarrow \sigma_t(\boldsymbol{x}) = g(\boldsymbol{x}) - \kappa(\boldsymbol{x}) u_t(\boldsymbol{x}). \tag{2.14}$$

Proof. (2.10) for $u_t(\boldsymbol{x}) = 0$ is equivalent to (2.12). For $u_t(\boldsymbol{x}) > 0$, (2.11) and (2.13) are equivalent, as (2.11) can be written by $\sigma_t(\boldsymbol{x}) u_t(\boldsymbol{x}) + g(\boldsymbol{x}) u_t(\boldsymbol{x}) + \kappa(\boldsymbol{x}) u_t^2(\boldsymbol{x}) = 0$ and we can divide by $u_t(\boldsymbol{x})$. Similarly, equivalence between (2.11) and (2.14) follows from the fact that in this case (2.11) is $\sigma_t(\boldsymbol{x}) u_t(\boldsymbol{x}) - g(\boldsymbol{x}) u_t(\boldsymbol{x}) + \kappa(\boldsymbol{x}) u_t^2(\boldsymbol{x}) = 0$ and we can divide by $u_t(\boldsymbol{x})$, again.□

The stick-slip condition for $d = 2$ is shown in Figure 1. Figure 1a indicates that a slip will not occur when $\sigma_t(\boldsymbol{x}) \in \langle -g(\boldsymbol{x}), g(\boldsymbol{x}) \rangle$. If $\sigma_t(\boldsymbol{x})$ will not be in this interval, then a slip will be occured and the relationship between $u_t(\boldsymbol{x})$ and stress $\sigma_t(\boldsymbol{x})$ is determined by the adhesion $\kappa(\boldsymbol{x})$. This stick-slip condition was first used in [1]. It is combination of the Tresca condition shown in Figure 1b [2] and the classical Navier condition shown in Figure 1c [3].



(a) $g > 0, \kappa > 0$ (b) $g > 0,\ \kappa = 0$ (c) $g = 0,\ \kappa > 0$

Figure 1: Stick-slip condition for different g and κ

The classic solution of the problem (2.1)-(2.7) are the functions $\boldsymbol{u} \in \left(C^2(\Omega)\right)^d$ and $p \in C^1(\Omega)$ with a continuous extension at the boundary $\partial\Omega$ such that all relations (2.1)-(2.7) are satisfied after substituting $\boldsymbol{u}$ and p.

2.2 Two-field weak formulation of the problem

The equation (2.1) can be write as follows:

$$-\nu\Delta u_i + \boldsymbol{u}\cdot\nabla u_i + p_{x_i} = f_i, \quad i = 1, \cdots, d, \tag{2.15}$$

where $\boldsymbol{u} = (u_1, \cdots, u_d)^\mathsf{T}$, $\nabla p_i = (p_{x_1}, \cdots, p_{x_d})^\mathsf{T}$, and $\boldsymbol{f} = (f_1, \cdots, f_d)^\mathsf{T}$. The middle term on the left side prescribes: $\boldsymbol{u}\cdot\nabla u_i = u_1 u_{ix_1} + \cdots + u_d u_{ix_d}$. We multiply equations (2.15) by the components of a sufficiently smooth test function $\boldsymbol{v} = (v_1, \cdots, v_d)^\mathsf{T}$, integrate them and use the Green formula. We will specify the requirements for the test function below.

From (2.15) we get:

$$-\nu\int_\Omega \Delta u_i v_i + \int_\Omega (\boldsymbol{u}\cdot\nabla u_i)v_i + \int_\Omega p_{x_i} v_i = \int_\Omega f_i v_i, \quad i = 1, \cdots, d. \tag{2.16}$$

We adjust the first and third integral in (2.16) by the Green formula, which we use in the following forms:

$$\int_\Omega \Delta u_i v_i = -\int_\Omega \nabla u_i\cdot\nabla v_i + \int_{\partial\Omega}\frac{\partial u_i}{\partial \boldsymbol{n}} v_i,$$
$$\int_\Omega p_{x_i} v_i = -\int_\Omega p v_{ix_i} + \int_{\partial\Omega} p v_i n_i,$$

where $\boldsymbol{n} = (n_1, \cdots, n_d)^\mathsf{T}$ is the unit vector of the outward normal to $\partial\Omega$ at $\boldsymbol{x} \in \partial\Omega$. We arrive at:

$$\nu\int_\Omega \nabla u_i\cdot\nabla v_i + \int_\Omega (\boldsymbol{u}\cdot\nabla u_i)v_i - \int_\Omega p v_{ix_i} = \int_\Omega f_i v_i + \int_{\partial\Omega}\nu\frac{\partial u_i}{\partial \boldsymbol{n}} v_i - p v_i n_i$$

for $i = 1, \cdots, d$. We summing this equations over i:

$$\nu\int_\Omega \nabla\boldsymbol{u} : \nabla\boldsymbol{v} + \int_\Omega (\boldsymbol{u}\cdot\nabla\boldsymbol{u})\cdot\boldsymbol{v} - \int_\Omega p(\nabla\cdot\boldsymbol{v}) = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v} + \int_{\partial\Omega}\boldsymbol{\sigma}\cdot\boldsymbol{v}, \tag{2.17}$$

where $\nabla \boldsymbol{u} : \nabla \boldsymbol{v} = \nabla u_1 \cdot \nabla v_1 + \cdots + \nabla u_d \cdot \nabla v_d$. We adjust the equation (2.17) using boundary conditions. First we divide the boundary integral into three parts:

$$\int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{v} = \int_{\gamma_D} \boldsymbol{\sigma} \cdot \boldsymbol{v} + \int_{\gamma_N} \boldsymbol{\sigma} \cdot \boldsymbol{v} + \int_{\gamma_S} \boldsymbol{\sigma} \cdot \boldsymbol{v}.$$

We will require that test function $\boldsymbol{v}$ satisfy the Dirichlet boundary condition $\boldsymbol{v} = \boldsymbol{u}_D$ on γ_D such that we write integral over γ_D as

$$\int_{\gamma_D} \boldsymbol{\sigma} \cdot \boldsymbol{v} = \int_{\gamma_D} \boldsymbol{\sigma} \cdot \boldsymbol{v}_D.$$

The integral over γ_N is determined by the Neumann boundary condition:

$$\int_{\gamma_N} \boldsymbol{\sigma} \cdot \boldsymbol{v} = \int_{\gamma_N} \boldsymbol{\sigma}_N \cdot \boldsymbol{v}.$$

Next, we divide the integral over γ_S into two parts:

$$\int_{\gamma_S} \boldsymbol{\sigma} \cdot \boldsymbol{v} = \int_{\gamma_S} \boldsymbol{\sigma}_t \cdot \boldsymbol{v}_t + \int_{\gamma_s} \sigma_n v_n,$$

where the second integral on the right site is equal to zero, because we will require that the non-penetration condition for the test functions be satisfied, i.e. $v_n = 0$.

These results will be written by the following forms:

$$\begin{aligned}
&a : \left(H^1(\Omega)\right)^d \times \left(H^1(\Omega)\right)^d \to \mathbb{R}, && a(\boldsymbol{w}, \boldsymbol{v}) = \nu \int_\Omega \nabla \boldsymbol{w} : \nabla \boldsymbol{v}, \\
&b : L^2(\Omega) \times \left(H^1(\Omega)\right)^d \to \mathbb{R}, && b(q, \boldsymbol{w}) = -\int_\Omega q(\nabla \cdot \boldsymbol{v}), \\
&c : \left(H^1(\Omega)\right)^d \times \left(H^1(\Omega)\right)^d \times \left(H^1(\Omega)\right)^d \to \mathbb{R}, && c(\boldsymbol{z}; \boldsymbol{w}, \boldsymbol{v}) = \int_\Omega (\boldsymbol{z} \cdot \nabla \boldsymbol{w}) \cdot \boldsymbol{v}, \\
&l : \left(H^1(\Omega)\right)^d \to \mathbb{R}, && l(\boldsymbol{v}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} + \int_{\gamma_N} \boldsymbol{\sigma}_N \cdot \boldsymbol{v}.
\end{aligned}$$

We define also the velocity solution set:

$$V_{\boldsymbol{u}_D} = \left\{ \boldsymbol{v} \in \left(H^1(\Omega)\right)^d : \boldsymbol{v} = \boldsymbol{u}_D \ \text{ on } \ \gamma_D, \ v_n = 0 \ \text{ on } \ \gamma_S \right\}$$

and by the same way we define the set of the test functions, $V = V_{\boldsymbol{u}_D}$. The equation (2.17) can be written for the $\boldsymbol{v} \in V_{\boldsymbol{u}_D}$ as follows:

$$a(\boldsymbol{u}, \boldsymbol{v}) + c(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{v}) + b(p, \boldsymbol{v}) = l(\boldsymbol{v}) + \int_{\gamma_D} \boldsymbol{\sigma} \cdot \boldsymbol{u}_D + \int_{\gamma_S} \boldsymbol{\sigma}_t \cdot \boldsymbol{v}_t. \tag{2.18}$$

We write the equation (2.18) for $\boldsymbol{v} = \boldsymbol{u}$:

$$a(\boldsymbol{u}, \boldsymbol{u}) + c(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{u}) + b(p, \boldsymbol{u}) = l(\boldsymbol{u}) + \int_{\gamma_D} \boldsymbol{\sigma} \cdot \boldsymbol{u}_D + \int_{\gamma_S} \boldsymbol{\sigma}_t \cdot \boldsymbol{u}_t. \tag{2.19}$$

By substracting (2.18) and (2.19) we get:

$$a(\boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}) + c(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}) + b(p, \boldsymbol{v} - \boldsymbol{u}) = l(\boldsymbol{v} - \boldsymbol{u}) + I, \tag{2.20}$$

where $I = \int_{\gamma_S} \boldsymbol{\sigma}_t \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t)$. Next, we will examine integral I. When we add and subtract some members into I we get:

$$I = \int_{\gamma_s} \boldsymbol{\sigma}_t \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t) + g(||\boldsymbol{v}_t|| - ||\boldsymbol{u}_t||) + \kappa \boldsymbol{u}_t \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t) - g(||\boldsymbol{v}_t|| - ||\boldsymbol{u}_t||) - \kappa \boldsymbol{u}_t \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t).$$

Using (2.7), we get:

$$I = \int_{\gamma_s} \boldsymbol{\sigma}_t \cdot \boldsymbol{v}_t + g||\boldsymbol{v}_t|| + \kappa \boldsymbol{u}_t \cdot \boldsymbol{v}_t - g(||\boldsymbol{v}_t|| - ||\boldsymbol{u}_t||) - \kappa \boldsymbol{u}_t \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t).$$

Using the Cauchy inequality and (2.6), we can show that the first three terms are nonnegative:

$$\boldsymbol{\sigma}_t \cdot \boldsymbol{v}_t + g||\boldsymbol{v}_t|| + \kappa \boldsymbol{u}_t \cdot \boldsymbol{v}_t = (\boldsymbol{\sigma}_t + \kappa \boldsymbol{u}_t) \cdot \boldsymbol{v}_t + g||\boldsymbol{v}_t|| \geq -||\boldsymbol{\sigma}_t + \kappa \boldsymbol{u}_t|| \cdot ||\boldsymbol{v}_t|| + g||\boldsymbol{v}_t|| \geq 0.$$

We have proved:

$$I \geq \int_{\gamma_s} -g(||\boldsymbol{v}_t|| - ||\boldsymbol{u}_t||) - \kappa \boldsymbol{u}_t \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t) = -j(\boldsymbol{v}, \boldsymbol{u}) + j(\boldsymbol{u}, \boldsymbol{u}), \tag{2.21}$$

where j is the sublinear form defined by:

$$j : \left(H^1(\Omega)\right)^d \times \left(H^1(\Omega)\right)^d \to \mathbb{R}, \quad j(\boldsymbol{v}, \boldsymbol{w}) = \int_{\gamma_S} g||\boldsymbol{v}_t|| + \kappa \boldsymbol{w}_t \cdot \boldsymbol{v}_t.$$

From (2.20) we get the variational inequality:

$$a(\boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}) + c(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}) + b(p, \boldsymbol{v} - \boldsymbol{u}) + j(\boldsymbol{v}, \boldsymbol{u}) - j(\boldsymbol{u}, \boldsymbol{u}) \geq l(\boldsymbol{v} - \boldsymbol{u}).$$

We also write the condition of incompressibility (2.2) in the variational way:

$$\int_{\Omega} q(\nabla \cdot \boldsymbol{u}) = 0,$$

where the set of the test functions is taken as $L^2(\Omega)$, i.e. $q \in L^2(\Omega)$.

We arrived at the following weak formulation of our problem (2.1)-(2.7):

$$\left.\begin{aligned} &\textit{Find } (\boldsymbol{u},p)\in V_{\boldsymbol{u}_D}\times L^2(\Omega) \textit{ such that for all } (\boldsymbol{v},q)\in V_{\boldsymbol{u}_D}\times L^2(\Omega):\\ &a(\boldsymbol{u},\boldsymbol{v}-\boldsymbol{u})+c(\boldsymbol{u};\boldsymbol{u},\boldsymbol{v}-\boldsymbol{u})+b(p,\boldsymbol{v}-\boldsymbol{u})+j(\boldsymbol{v},\boldsymbol{u})-j(\boldsymbol{u},\boldsymbol{u})\geq l(\boldsymbol{v}-\boldsymbol{u}),\\ &b(q,\boldsymbol{u})=0. \end{aligned}\right\} \tag{2.22}$$

The solution to (2.22) is called the weak solution of (2.1)-(2.7). The following existence therorem have been proved in [1] for $\boldsymbol{u}_D=\boldsymbol{0}$, $\gamma_N=\emptyset$, and $\kappa=\text{const}$.

Theorem 1 *Let $\boldsymbol{f}\in(L^2(\Omega))^d$, $\gamma_N=\emptyset$, $g\in L^2(\gamma_S)$, $g\geq 0$. The problem (2.22) has a unique solution, if*

$$0<\frac{C_cC_1(||\boldsymbol{f}||^*+||g||_{L^2(\gamma_S)})}{\nu^2}<1,$$
$$0<\frac{C_0\kappa}{2\nu}<1-\frac{C_cC_1(||\boldsymbol{f}||^*+||g||_{L^2(\gamma_S)})}{2\nu^2},$$

where $||\cdot||^*$ is the dual norm, C_c and C_0 are the boundness constants for the trilinear form c and the sublinear form j, respectively, and $C_1>0$ (see. [1]).

It is proved in [5] that the problem (2.22) without the trilinear form c and the problem (2.1)-(2.7) without the convective term are equivalent, if the solution is sufficiently smooth. The proof may be easily adaptet for our case.

2.3 Four-field formulation of the problem

In this case we defined the velocity solution set as:

$$V_{\boldsymbol{u}_D}=\left\{\boldsymbol{v}\in\left(H^1(\Omega)\right)^d:\boldsymbol{v}=\boldsymbol{u}_D \text{ on } \gamma_D\right\}$$

and also we choose the space of the test functions as $V_{\boldsymbol{0}}$. The equation (2.17) can be written for $\boldsymbol{v}\in V_{\boldsymbol{0}}$ as:

$$a(\boldsymbol{u},\boldsymbol{v})+c(\boldsymbol{u};\boldsymbol{u},\boldsymbol{v})+b(p,\boldsymbol{v})-\int_{\gamma_S}\sigma_n v-\int_{\gamma_S}\boldsymbol{\sigma}_t\cdot\boldsymbol{v}_t=l(\boldsymbol{v}).$$

We will also use the inequality (2.21):

$$\int_{\gamma_S}\boldsymbol{\sigma}_t\cdot(\boldsymbol{v}_t-\boldsymbol{u}_t)+j(\boldsymbol{v},\boldsymbol{u})-j(\boldsymbol{u},\boldsymbol{u})\geq 0.$$

Finally, the condition of incompressibility and the condition of non-penetrability are written in the variational way:

$$\int_\Omega q(\nabla\cdot\boldsymbol{u})=0,\quad q\in L^2(\Omega),$$
$$\int_{\gamma_S}\varphi u_n=0,\quad \varphi\in L^2(\gamma_S),$$

respectively. Summarizing these results, we obtain the following weak formulation of the problem (2.1)-(2.7):

$$\left.\begin{aligned}
&\textit{Find } (\boldsymbol{u}, p, \sigma_n, \boldsymbol{\sigma}_t) \in V_{\boldsymbol{u}_D} \times L^2(\Omega) \times L^2(\gamma_S) \times \left(L^2(\gamma_S)\right)^{d-1} \textit{ such that} \\
&a(\boldsymbol{u}, \boldsymbol{v}) + c(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{v}) + b(p, \boldsymbol{v}) - \int_{\gamma_S} \sigma_n v_n - \int_{\gamma_S} \boldsymbol{\sigma}_t \cdot \boldsymbol{v}_t = l(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in V_{\mathbf{0}}, \\
&b(q, \boldsymbol{u}) = 0 \quad \forall q \in L^2(\Omega), \\
&\int_{\gamma_S} \varphi u_n = 0 \quad \forall \varphi \in L^2(\gamma_S), \\
&\int_{\gamma_t} \boldsymbol{\sigma}_t \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t) + j(\boldsymbol{v}, \boldsymbol{u}) - j(\boldsymbol{u}, \boldsymbol{u}) \geq 0 \quad \forall \boldsymbol{v} \in V_{\boldsymbol{u}_D}.
\end{aligned}\right\} \tag{2.23}$$

The assumptions of the existence theorem (1) guarantee also the solvability of the problem (2.23). The following theorem proves the equivalence of the problem (2.23) and our original problem (2.1)-(2.7).

Theorem 2 *a) Every classical solution to (2.1)-(2.7) is a solution to (2.23), as well.*
b) If the solution to (2.23) is sufficiently smooth, then it solves (2.1)-(2.7).

Proof. The statement (a) is a consequence of the previous construction. We can prove the statement (b). The first variational equation from (2.23) can be written as follows using Green's theorem:

$$\int_\Omega (-\nu \Delta \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p) \cdot \boldsymbol{v} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} + \int_{\gamma_N} \boldsymbol{\sigma}_N \cdot \boldsymbol{v} - \int_{\gamma_D \cup \gamma_N} \boldsymbol{\sigma} \cdot \boldsymbol{v}. \tag{2.24}$$

We will choose $\boldsymbol{v} = \boldsymbol{\psi} \in (C_0^\infty(\Omega))^d$ so that the boundary integrals are equal to zero. Therefore we arrive at

$$\int_\Omega (-\nu \Delta \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p - \boldsymbol{f}) \cdot \boldsymbol{\psi} = 0 \quad \forall \boldsymbol{\psi} \in (C_0^\infty(\Omega))^d$$

and from this variational equation it follows (2.1). If we use this result in (2.24), we get

$$\int_{\gamma_N} (\boldsymbol{\sigma}_N - \boldsymbol{\sigma}) \cdot \boldsymbol{v} = 0 \quad \forall \boldsymbol{v} \in V_0,$$

which implies (2.4). The equations (2.3) and (2.5) follow directly from the second and the third variational equation in (2.23). Finally, it remains to prove that the stick-slip boundary conditions (2.6), (2.7) are satisfied. It is prooved in [4] using the Hahn-Banach theorem, that from the variational inequality in (2.23) it follows:

$$||\boldsymbol{\sigma}_t + \kappa \boldsymbol{u}_t|| \leq g \quad \textit{on } \gamma_S, \tag{2.25}$$

i.e. (2.6) holds. For $\boldsymbol{v} = \boldsymbol{0}$ and $\boldsymbol{v} = 2\boldsymbol{u}$ we get from the variational inequality at (2.23):

$$\int_{\gamma_S} \boldsymbol{\sigma}_t \cdot \boldsymbol{u}_t + g||\boldsymbol{u}_t|| + \kappa \boldsymbol{u}_t \cdot \boldsymbol{u}_t \leq 0$$

and

$$\int_{\gamma_S} \boldsymbol{\sigma}_t \cdot \boldsymbol{u}_t + g||\boldsymbol{u}_t|| + \kappa \boldsymbol{u}_t \cdot \boldsymbol{u}_t \geq 0,$$

respectively. Therefore

$$\int_{\gamma_S} \boldsymbol{\sigma}_t \cdot \boldsymbol{u}_t + g||\boldsymbol{u}_t|| + \kappa \boldsymbol{u}_t \cdot \boldsymbol{u}_t = 0. \tag{2.26}$$

Using the Cauchy inequality and (2.25) we get:

$$(\boldsymbol{\sigma}_t + \kappa \boldsymbol{u}_t) \cdot \boldsymbol{u}_t + g||\boldsymbol{u}_t|| \geq -||\boldsymbol{\sigma}_t + \kappa \boldsymbol{u}_t|| \cdot ||\boldsymbol{u}_t|| + g||\boldsymbol{u}_t|| \geq 0.$$

It yields

$$\boldsymbol{\sigma}_t \cdot \boldsymbol{u}_t + g||\boldsymbol{u}_t|| + \kappa \boldsymbol{u}_t \cdot \boldsymbol{u}_t = 0,$$

that is (2.7). □

2.4 Ossen iterations in the continuous case

Our problem contains two nonlinearities: the convective term and the stick-slip boundary condition. We treat the second of these nonlinearities in the discrete case using the semi-smooth Newton method. Now, in the continuous case, we use the Ossen (Picard) iterations that linearize the convective term. This iterations generate a sequence of approximations from given $\boldsymbol{u}^0 \in V_{\boldsymbol{u}_D}$

$$(\boldsymbol{u}^1, p^1, \sigma_n^1, \boldsymbol{\sigma}_t^1), (\boldsymbol{u}^2, p^2, \sigma_n^2, \boldsymbol{\sigma}_t^2) \ldots \in V_{\boldsymbol{u}_D} \times L^2(\Omega) \times L^2(\gamma_S) \times \left(L^2(\gamma_S)\right)^{d-1}$$

that converges to the solution of the problem (2.23). The Ossen iterations represent the following iterative process:

$$\left.\begin{array}{l} \textit{Choose } \boldsymbol{u}^0 \in V_{\boldsymbol{u}_D}. \\ \textit{Find } (\boldsymbol{u}^k, p^k, \sigma_n^k, \boldsymbol{\sigma}_t^k) \in V_{\boldsymbol{u}_D} \times L^2(\Omega) \times L^2(\gamma_S) \times \left(L^2(\gamma_S)\right)^{d-1} \textit{ for } k \geq 1 \textit{ such that} \\ a(\boldsymbol{u}^k, \boldsymbol{v}) + c(\boldsymbol{u}^{k-1}; \boldsymbol{u}^k, \boldsymbol{v}) + b(p^k, \boldsymbol{v}) - \int_{\gamma_S} \sigma_n^k v_n - \int_{\gamma_S} \boldsymbol{\sigma}_t^k \cdot \boldsymbol{v}_t = l(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in V_{\boldsymbol{0}}, \\ b(q, \boldsymbol{u}^k) = 0 \quad \forall q \in L^2(\Omega), \\ \int_{\gamma_S} \varphi u_n^k = 0 \quad \forall \varphi \in L^2(\gamma_S), \\ \int_{\gamma_S} \boldsymbol{\sigma}_t^k \cdot (\boldsymbol{v}_t - \boldsymbol{u}_t^k) + j(\boldsymbol{v}, \boldsymbol{u}^k) - j(\boldsymbol{u}^k, \boldsymbol{u}^k) \geq 0 \quad \forall \boldsymbol{v} \in V_{\boldsymbol{u}_D}. \end{array}\right\} \tag{2.27}$$

It is proved in [1] that the Ossen iterations generate a sequence converging to the solution of the problem (2.23), if the assumptions of Theorem 1 are satisfied. The linearized problem (2.27), which we solve in each step of the Ossen iterations, will be called the Ossen problem (with the stick-slip condition). We will approximate this problem using the mixed finite element method in the following section.

3 Discretization

The Ossen problem (2.27) will be approximated using the mixed finite element method. We use the P1-bubble/P1 finite element pair introduced by Arnold, Brezzi, and Fortin [6]. This pair satisfies the inf-sup stability condition that has good approximation properties and a small degree of freedom. We will assembly stiffness matrices using vectorized codes proposed by J. Koko in [7], extended in [8] and finally described for our case with a convective term for two and three dimension in the Appendix A-C.

3.1 Mixed finite element method

Let $\mathcal{T}^h$ be a regular partition of $\overline{\Omega}$ and $T \in \mathcal{T}^h$ be its segment. We will assume that T is a triangle for $d = 2$ and a tetrahedron for $d = 3$ with vertices $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1}$ (with the local indices on T). We associate the local linear basis function $\phi_i^{(T)}(\boldsymbol{x})$ to each vertex of T so that $\phi_i^{(T)}(\boldsymbol{x}_j) = \delta_{ij},\ i,j = 1,\ldots,d+1$. The local bubble function on T is defined by $\phi_b^{(T)}(\boldsymbol{x}) = (d+1)^{d+1}\phi_1^{(T)}(\boldsymbol{x})\ldots\phi_{d+1}^{(T)}(\boldsymbol{x}),\ \boldsymbol{x} \in T$. Note that $\phi_b^{(T)}(\boldsymbol{x}) = 0$ on boundary of T. Next we define sets of functions:

$$
\begin{aligned}
B^h &= \left\{ v^h \in C(\overline{\Omega}) : v^h_{|_T} = c^{(T)}\phi_b^{(T)},\ c^{(T)} \in \mathbb{R} \quad \forall T \in \mathcal{T}^h \right\}, \\
W^h &= \left\{ v^h \in C(\overline{\Omega}) : v^h_{|_T} \in P^1(T) \quad \forall T \in \mathcal{T}^h \right\}, \\
W^h_{\gamma_S} &= \left\{ \varphi^h \in C(\overline{\gamma_S}) : \varphi^h = v^h_{|_{\overline{\gamma_S}}}, \quad v^h \in W^h \right\}, \\
V^h &= \left(W^h \oplus B^h \right)^d, \\
V^h_{\boldsymbol{u}_D} &= \left\{ \boldsymbol{v}^h \in V^h : \boldsymbol{v}^h(\boldsymbol{x}_i) = \boldsymbol{u}_D(\boldsymbol{x}_i) \quad \forall \boldsymbol{x}_i \in \overline{\gamma_D} \right\},
\end{aligned}
$$

where $\boldsymbol{x}_i \in \Omega,\ 1 \le i \le n$, are nodes of $\mathcal{T}^h$ (with the global indices on $\overline{\Omega}$).

In the discrete formulation of the Ossen problem (2.27) we change the sign of the stress on γ_S, i.e. we use the substitution $s_n = -\sigma_n$ and $\boldsymbol{s}_t = -\boldsymbol{\sigma}_t$. Next, we denote by $\boldsymbol{w}^h \in V_{\boldsymbol{u}_D}$ the aproximation of $\boldsymbol{u}^{k-1}$ from (2.27). The mixed finite element aproximation of the Ossen problem without the iterative index k reads as follows:

$$
\left.
\begin{aligned}
&\textit{Find } (\boldsymbol{u}^h, p^h, s_n^h, \boldsymbol{s}_t^h) \in V^h_{\boldsymbol{u}_D} \times W^h \times W^h_{\gamma_S} \times \left(W^h_{\gamma_S} \right)^{d-1} \ \textit{such that} \\
&a(\boldsymbol{u}^h, \boldsymbol{v}^h) + c(\boldsymbol{w}^h; \boldsymbol{u}^h, \boldsymbol{v}^h) + b(p^h, \boldsymbol{v}^h) + I_1^h(s_n^h, v_n^h) + I_2^h(\boldsymbol{s}_t^h, \boldsymbol{v}_t^h) = l(\boldsymbol{v}^h) \quad \forall \boldsymbol{v}^h \in V^h_{\boldsymbol{0}}, \\
&b(q^h, \boldsymbol{u}^h) = 0 \quad \forall q^h \in W^h, \\
&I_3^h(\varphi^h, u_n^h) = 0 \quad \forall \varphi^h \in W^h_{\gamma_S}, \\
&I_4^h(\boldsymbol{s}_t^h, \boldsymbol{u}_t^h, \boldsymbol{v}_t^h) + j^h(\boldsymbol{v}^h, \boldsymbol{u}^h) - j^h(\boldsymbol{u}^h, \boldsymbol{u}^h) \ge 0 \quad \forall \boldsymbol{v}^h \in V^h_{\boldsymbol{u}_D},
\end{aligned}
\right\} \tag{3.1}
$$

where $I_1^h, \ldots, I_4^h$, and j^h are approximations of the corresponding integrals from (2.27) derived by numerical integrations.

Let n be the number of nodes of the partition $\mathcal{T}^h$, n_D is the number of nodes lying on $\overline{\gamma_D}$, and n_S is the number of nodes lying on $\overline{\gamma_S}\backslash\overline{\gamma_D}$. In the algebraic counterpart of the problem (3.1) we take into acount only nodal values $\boldsymbol{u}^h(\boldsymbol{x}_i),\ \boldsymbol{x}_i \notin \overline{\gamma_D}$, since the bubble-coefficients are eliminated on the local level during the assembling procedure:

$$\mathbf{u} = \left(\boldsymbol{u}^h(\boldsymbol{x}_1)^\mathsf{T}, \boldsymbol{u}^h(\boldsymbol{x}_2)^\mathsf{T}, \ldots, \boldsymbol{u}^h(\boldsymbol{x}_{n-n_D})^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^{n_u}, \quad n_u = d(n - n_D),$$

where we use the global node numbering on $\overline{\Omega}$. The vectors $\mathbf{w},\ \mathbf{v} \in \mathbb{R}^{n_u}$ have similar meaning with respect to the functions $\boldsymbol{w}^h,\ \boldsymbol{v}^h$. Also the algebraic pressure component will be represented similarly:

$$\mathbf{p} = \left(p^h(\boldsymbol{x}_1), p^h(\boldsymbol{x}_2), \ldots, p^h(\boldsymbol{x}_{n_p})\right)^\mathsf{T} \in \mathbb{R}^{n_p}, \quad n_p = n.$$

The stress we will represent by the nodal values $s_n^h(\boldsymbol{x}_i),\ \boldsymbol{s}_t^h(\boldsymbol{x}_i)$ multiplied by the measure $\mu(\boldsymbol{x}_i)$ of the element corresponding to the node $\boldsymbol{x}_i \in \overline{\gamma_S}$:

$$\mathbf{s}_n = (s_{n,1}, s_{n,2}, \ldots, s_{n,n_S})^\mathsf{T} \in \mathbb{R}^{n_S}, \quad s_{n,i} = \mu(\boldsymbol{x}_i) s_n^h(\boldsymbol{x}_i),$$

for $d = 2$:

$$\mathbf{s}_t = (s_{t,1}, s_{t,2}, \ldots, s_{t,n_S})^\mathsf{T} \in \mathbb{R}^{n_S}, \quad s_{t,i} = \mu(\boldsymbol{x}_i) \boldsymbol{s}_t^h(\boldsymbol{x}_i),$$

and for $d = 3$:

$$\begin{aligned}
\mathbf{s}_t &= \left(\boldsymbol{s}_{t_1}^\mathsf{T}, \boldsymbol{s}_{t_2}^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^{2n_S}, \\
\boldsymbol{s}_{t_j} &= \left(s_{t_j,1}, s_{t_j,2}, \ldots, s_{t_j,n_S}\right)^\mathsf{T} \in \mathbb{R}^{n_S}, \quad s_{t_j,i} = \mu(\boldsymbol{x}_i) \boldsymbol{s}_{t_j}^h(\boldsymbol{x}_i),\ j = 1, 2,
\end{aligned}$$

where we use the local node numbering $\boldsymbol{x}_i$ on $\overline{\gamma_S}$. The algebraic form of the stress components result from the numerical integration of the respective integral:

$$I_1^h(s_n^h, v_n^h) = \sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i) s_n^h(\boldsymbol{x}_i) \left(\boldsymbol{n}(\boldsymbol{x}_i) \cdot \boldsymbol{v}^h(\boldsymbol{x}_i)\right) = \mathbf{v}^\mathsf{T} \mathbf{N}^\mathsf{T} \mathbf{s}_n,$$

where the *i-th* row of $\mathbf{N} \in \mathbb{R}^{n_S \times n_u}$ has nonzero elements given by the vector $\boldsymbol{n}(\boldsymbol{x}_i) \in \mathbb{R}^d$ on suitable positions. Analogously we get for $d = 2$:

$$I_2^h(\boldsymbol{s}_t^h, \boldsymbol{v}_t^h) = \sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i) \boldsymbol{s}_t^h(\boldsymbol{x}_i) \left(\boldsymbol{t}(\boldsymbol{x}_i) \cdot \boldsymbol{v}^h(\boldsymbol{x}_i)\right) = \mathbf{v}^\mathsf{T} \mathbf{T}^\mathsf{T} \mathbf{s}_n,$$

where the *i-th* row of $\mathbf{T} \in \mathbb{R}^{n_S \times n_u}$ is given by $\boldsymbol{t}(\boldsymbol{x}_i) \in \mathbb{R}^2$; and for $d = 3$:

$$\begin{aligned}
I_2^h(\boldsymbol{s}_t^h, \boldsymbol{v}_t^h) &= \sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i) \left(\boldsymbol{s}_{t_1}^h(\boldsymbol{x}_i)(\boldsymbol{t}_1(\boldsymbol{x}_i) \cdot \boldsymbol{v}^h(\boldsymbol{x}_i)) + \boldsymbol{s}_{t_2}^h(\boldsymbol{x}_i)(\boldsymbol{t}_2(\boldsymbol{x}_i) \cdot \boldsymbol{v}^h(\boldsymbol{x}_i))\right) = \\
&= \mathbf{v}^\mathsf{T} \mathbf{T}_1^\mathsf{T} \mathbf{s}_{t_1} + \mathbf{v}^\mathsf{T} \mathbf{T}_2^\mathsf{T} \mathbf{s}_{t_2} = \mathbf{v}^\mathsf{T} \mathbf{T}^\mathsf{T} \mathbf{s}_t,
\end{aligned}$$

where the *i-th* row of $\mathbf{T}_1, \mathbf{T}_2 \in \mathbb{R}^{n_S \times n_u}$ are given by $\boldsymbol{t}_1(\boldsymbol{x}_i), \boldsymbol{t}_2(\boldsymbol{x}_i)$, respectively, and $\mathbf{T} = \left(\mathbf{T}_1^\mathsf{T}, \mathbf{T}_2^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^{2n_S \times n_u}$. The integral representing the non-penetration condition gives:

$$I_3^h(\varphi^h, u_n^h) = \sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i)\varphi^h(\boldsymbol{x}_i)\left(\boldsymbol{n}(\boldsymbol{x}_i) \cdot \boldsymbol{u}^h(\boldsymbol{x}_i)\right) = \boldsymbol{\varphi}^\mathsf{T}\mathbf{N}^\mathsf{T}\mathbf{u}_n, \quad \boldsymbol{\varphi} \in \mathbb{R}^{n_S}.$$

It remains to approximate the integrals from the variational inequality describing the stick-slip condition. For $d = 2$ we get:

$$\begin{aligned} I_4^h(\boldsymbol{s}_t^h, \boldsymbol{u}_t^h, \boldsymbol{v}_t^h) &= -\sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i)\boldsymbol{s}_t^h(\boldsymbol{x}_i)\left(\boldsymbol{t}(\boldsymbol{x}_i) \cdot (\boldsymbol{v}^h(\boldsymbol{x}_i) - \boldsymbol{u}^h(\boldsymbol{x}_i))\right) \\ &= -\sum_{i=1}^{n_S} \boldsymbol{s}_{t,i}\left((\mathbf{T}\mathbf{v})_i - (\mathbf{T}\mathbf{u})_i\right) \\ &= -\mathbf{s}_t^\mathsf{T}\mathbf{T}(\mathbf{v} - \mathbf{u}) \end{aligned}$$

and

$$\begin{aligned} j^h(\boldsymbol{u}^h, \boldsymbol{v}^h) &= \sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i)\left(g(\boldsymbol{x}_i)|v_t^h(\boldsymbol{x}_i)| + \kappa(\boldsymbol{x}_i)u_t^h(\boldsymbol{x}_i)v_t^h(\boldsymbol{x}_i)\right) \\ &= \sum_{i=1}^{n_S} g_i|(\mathbf{T}\mathbf{v})_i| + \kappa_i(\mathbf{T}\mathbf{u})_i(\mathbf{T}\mathbf{v})_i \\ &= \mathbf{g}^\mathsf{T}|\mathbf{T}\mathbf{v}| + \mathbf{v}^\mathsf{T}\mathbf{T}^\mathsf{T}\mathbf{D}(\boldsymbol{\kappa})\mathbf{T}\mathbf{u}, \end{aligned}$$

where $\mathbf{g} = (g_1, \ldots, g_{n_S})^\mathsf{T} \in \mathbb{R}^{n_S}$, $g_i = \mu(\mathbf{x}_i)g(\mathbf{x}_i)$, $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_{n_S})^\mathsf{T} \in \mathbb{R}^{n_S}$, $\kappa_i = \mu(\mathbf{x}_i)\kappa(\mathbf{x}_i)$, $\mathbf{D}(\boldsymbol{\kappa}) = diag(\boldsymbol{\kappa}) \in \mathbb{R}^{n_s \times n_s}$, and the absolute value applied to the vector is interpreted component wisely.

For $d = 3$ we denote:

$$\left.\begin{aligned} \mathbf{s}_{t,i} &= (s_{t_1,i}, s_{t_2,i})^\mathsf{T} \in \mathbb{R}^2, \\ \mathbf{v}_{t,i} &= (\mathbf{T}_1\mathbf{v})_i, (\mathbf{T}_2\mathbf{v})_i)^\mathsf{T} \in \mathbb{R}^2, \\ \mathbf{u}_{t,i} &= ((\mathbf{T}_1\mathbf{u})_i, (\mathbf{T}_2\mathbf{u})_i)^\mathsf{T} \in \mathbb{R}^2 \end{aligned}\right\} \tag{3.2}$$

for $i = 1, \ldots, n_S$. Then the integrals that describe the stick-slip condition can be written as follows:

$$\begin{aligned} I_4^h(\boldsymbol{s}_t^h, \boldsymbol{u}_t^h, \boldsymbol{v}_t^h) &= -\sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i)\left(s_{t_1}^h(\boldsymbol{x}_i)\left(\boldsymbol{t}_1(\boldsymbol{x}_i) \cdot (\boldsymbol{v}^h(\boldsymbol{x}_i) - \boldsymbol{u}^h(\boldsymbol{x}_i))\right) + \right. \\ &\qquad \left. + s_{t_2}^h(\boldsymbol{x}_i)\left(\boldsymbol{t}_2(\boldsymbol{x}_i) \cdot (\boldsymbol{v}^h(\boldsymbol{x}_i) - \boldsymbol{u}^h(\boldsymbol{x}_i))\right)\right) \\ &= -\sum_{i=1}^{n_S} \mathbf{s}_{t,i}^\mathsf{T}\left(\mathbf{v}_{t,i} - \mathbf{u}_{t,i}\right) \end{aligned}$$

and

$$j^h(\boldsymbol{u}^h, \boldsymbol{v}^h) = \sum_{i=1}^{n_S} \mu(\boldsymbol{x}_i) \left(g(\boldsymbol{x}_i) ||\boldsymbol{v}_t^h(\boldsymbol{x}_i)|| + \kappa(\boldsymbol{x}_i) \boldsymbol{u}_t^h(\boldsymbol{x}_i) \cdot \boldsymbol{v}_t^h(\boldsymbol{x}_i) \right)$$
$$= \sum_{i=1}^{n_S} g_i ||\mathbf{v}_{t,i}|| + \kappa_i \mathbf{u}_{t,i}^{\mathsf{T}} \mathbf{v}_{t,i},$$

where the meaning of g_i and κ_i, $i = 1, \ldots, n_S$, is formally the same as for $d = 2$.

3.2 Algebraic Ossen problem for $d = 2$

Summarizing the previous results for $d = 2$, we arrive at the following algebraic form of the Ossen problem:

$$\left.\begin{aligned} &\textit{Find } (\mathbf{u}, \mathbf{p}, \mathbf{s}_n, \mathbf{s}_t) \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_S} \times \mathbb{R}^{n_S} \textit{ such that} \\ &\mathbf{A}(\mathbf{w})\mathbf{u} + \mathbf{B}_u^{\mathsf{T}}(\mathbf{w})\mathbf{p} + \mathbf{N}^{\mathsf{T}}\mathbf{s}_n + \mathbf{T}^{\mathsf{T}}\mathbf{s}_t = \mathbf{b}(\mathbf{w}), \\ &\mathbf{B}_l(\mathbf{w})\mathbf{u} - \mathbf{E}(\mathbf{w})\mathbf{p} = \mathbf{c}(\mathbf{w}), \\ &\mathbf{N}\mathbf{u} = \mathbf{0}, \\ &-\mathbf{s}_t^{\mathsf{T}}\mathbf{T}(\mathbf{v} - \mathbf{u}) + \mathbf{g}^{\mathsf{T}}\left(|\mathbf{T}\mathbf{v}| - |\mathbf{T}\mathbf{u}|\right) + \mathbf{u}^{\mathsf{T}}\mathbf{T}^{\mathsf{T}}\mathbf{D}(\boldsymbol{\kappa})\mathbf{T}(\mathbf{v} - \mathbf{u}) \geq 0 \quad \forall \mathbf{v} \in \mathbb{R}^{n_u}, \end{aligned}\right\} \tag{3.3}$$

where $\mathbf{w} \in \mathbb{R}^{n_u}$ is given. Here, $\mathbf{A}(\mathbf{w}) \in \mathbb{R}^{n_u \times n_u}$ is the nonsymmetric matrix representing the diffusion and convective term, $\mathbf{B}_u(\mathbf{w}), \mathbf{B}_l(\mathbf{w}) \in \mathbb{R}^{n_p \times n_u}$ are the full row rank matrices representing the gradient and divergence term, $\mathbf{E}(\mathbf{w}) \in \mathbb{R}^{n_p \times n_p}$ is regular matrix arising from the elimination of the bubble coefficients, and $\mathbf{b}(\mathbf{w}) \in \mathbb{R}^{n_u}, \mathbf{c}(\mathbf{w}) \in \mathbb{R}^{n_p}$ are the right hand side vector. Let us notice that $\mathbf{w}$ representing $\boldsymbol{w}^h$ from the convective term appears in all objects above due to the elimination of the buble coefficients. Next, $\mathbf{N}, \mathbf{T} \in \mathbb{R}^{n_S \times n_u}$ and $\mathbf{g}, \boldsymbol{\kappa} \in \mathbb{R}_+^{n_S}$. This objects do not depend on $\mathbf{w}$, as they are created from the boundary integrals on γ_S and on the boundary are the bubble functions equal to zero.

In the following lemma we prescribe the variationa inequality from (3.3) into a form that is suitable for the calculation.

Lemma 2 *The variational inequality*

$$-\mathbf{s}_t^{\mathsf{T}}\mathbf{T}(\mathbf{v} - \mathbf{u}) + \mathbf{g}^{\mathsf{T}}\left(|\mathbf{T}\mathbf{v}| - |\mathbf{T}\mathbf{u}|\right) + \mathbf{u}^{\mathsf{T}}\mathbf{T}^{\mathsf{T}}\mathbf{D}(\boldsymbol{\kappa})\mathbf{T}(\mathbf{v} - \mathbf{u}) \geq 0 \quad \forall \mathbf{v} \in \mathbb{R}^{n_S}$$

is satisfied iff the following relations are hold

$$\left.\begin{aligned} &(\mathbf{T}\mathbf{u})_i = 0 \Rightarrow |s_{ti}| \leq g_i, \\ &(\mathbf{T}\mathbf{u})_i > 0 \Rightarrow s_{ti} = g_i + \kappa_i (\mathbf{T}\mathbf{u})_i, \\ &(\mathbf{T}\mathbf{u})_i < 0 \Rightarrow s_{ti} = -g_i + \kappa_i (\mathbf{T}\mathbf{u})_i, \end{aligned}\right\} i \in \mathcal{N} = \{1, \ldots, n_S\}.$$

Proof. We prove the implication "⇒". We write the variational inequality as follows:

$$\sum_{j=1}^{n_S} -s_{t,j}\left((\mathbf{Tv})_j - (\mathbf{Tu})_j\right) + g_j\left(|(\mathbf{Tv})_j| - |(\mathbf{Tu})_j|\right) + \kappa_j(\mathbf{Tu})_j\left((\mathbf{Tv})_j - (\mathbf{Tu})_j\right) \geq 0 \quad \forall \mathbf{v} \in \mathbb{R}^{n_u}. \tag{3.4}$$

We choose:

$$\mathbf{v}^{+\varepsilon} \in \mathbb{R}^{n_u} : (\mathbf{Tv}^{+\varepsilon})_j = (\mathbf{Tu})_j,\ j \neq i \ \text{ and } \ (\mathbf{Tv}^{+\varepsilon})_i = (\mathbf{Tu})_i + \varepsilon,$$
$$\mathbf{v}^{-\varepsilon} \in \mathbb{R}^{n_u} : (\mathbf{Tv}^{-\varepsilon})_j = (\mathbf{Tu})_j,\ j \neq i \ \text{ and } \ (\mathbf{Tv}^{-\varepsilon})_i = (\mathbf{Tu})_i - \varepsilon,$$

where $\varepsilon > 0$. For $\mathbf{v}^{+\varepsilon}$ and $\mathbf{v}^{-\varepsilon}$ we get from (3.4) the following inequalities:

$$-s_{t,i}\varepsilon + g_i(|(\mathbf{Tu})_i + \varepsilon| - |(\mathbf{Tu})_i|) + \kappa_i(\mathbf{Tu})_i\varepsilon \geq 0,$$

and

$$s_{t,i}\varepsilon + g_i(|(\mathbf{Tu})_i - \varepsilon| - |(\mathbf{Tu})_i|) - \kappa_i(\mathbf{Tu})_i\varepsilon \geq 0,$$

respectively. If $(\mathbf{Tu})_i = 0$, we get:

$$\left.\begin{aligned} -s_{t,i}\varepsilon + g_i\varepsilon \geq 0, \\ s_{t,i}\varepsilon + g_i\varepsilon \geq 0, \end{aligned}\right\} \quad |s_{t,i}| \leq g_i.$$

If $(\mathbf{Tu})_i > 0$ and ε is sufficiently small, we get:

$$\left.\begin{aligned} -s_{t,i}\varepsilon + g_i\varepsilon + \kappa_i(\mathbf{Tu})_i\varepsilon \geq 0, \\ s_{t,i}\varepsilon - g_i\varepsilon - \kappa_i(\mathbf{Tu})_i\varepsilon \geq 0, \end{aligned}\right\} \quad s_{t,i} = g_i + \kappa_i(\mathbf{Tu})_i.$$

For $(\mathbf{Tu})_i < 0$, the proof is analogous. The implication "⇐" can be proved easily, but it is not needed for this paper. □

3.3 Algebraic Ossen problem for $d = 3$

Summarizing the previous results for $d = 3$, we arrive at the following algebraic form of the Ossen problem:

$$\left.\begin{aligned} &\textit{Find } (\mathbf{u}, \mathbf{p}, \mathbf{s}_n, \mathbf{s}_t) \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_S} \times \mathbb{R}^{2n_S} \textit{ such that} \\ &\mathbf{A}(\mathbf{w})\mathbf{u} + \mathbf{B}_u^{\mathsf{T}}(\mathbf{w})\mathbf{p} + \mathbf{N}^{\mathsf{T}}\mathbf{s}_n + \mathbf{T}^{\mathsf{T}}\mathbf{s}_t = \mathbf{b}(\mathbf{w}), \\ &\mathbf{B}_l(\mathbf{w})\mathbf{u} - \mathbf{E}(\mathbf{w})\mathbf{p} = \mathbf{c}(\mathbf{w}), \\ &\mathbf{Nu} = \mathbf{0}, \\ &-\sum_{j=1}^{n_S} \mathbf{s}_{t,j}^{\mathsf{T}}(\mathbf{v}_{t,j} - \mathbf{u}_{t,j}) + g_j(||\mathbf{v}_{t,j}|| - ||\mathbf{u}_{t,j}||) + \kappa_j\mathbf{u}_{t,j}^{\mathsf{T}}(\mathbf{v}_{t,j} - \mathbf{u}_{t,j}) \geq 0 \quad \forall \mathbf{v} \in \mathbb{R}^{n_u}, \end{aligned}\right\} \tag{3.5}$$

where $\mathbf{w} \in \mathbb{R}^{n_u}$ is given. The formal meaning of the most of object in (3.5) is the same as for $d = 2$. The differences lie in

$$\mathbf{s}_t = \begin{pmatrix} \mathbf{s}_{t_1} \\ \mathbf{s}_{t_2} \end{pmatrix} \in \mathbb{R}^{2n_S}, \quad \mathbf{T} = \begin{pmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{pmatrix} \in \mathbb{R}^{2n_S \times n_u},$$

in the definitions (3.2), and in the fact that we use the Euclidean norm $||\cdot||$ in $\mathbb{R}^2$ instead of the absolute value.

In the following lemma we write the variation inequality from (3.5) in two equivalent ways. We will use the projection on a circle in $\mathbb{R}^2$.

Definition 1 *Let $C(r) = \{\boldsymbol{x} \in \mathbb{R}^2 : ||\boldsymbol{x}|| \leq r\}$ be a circle in $\mathbb{R}^2$ with the radius r. The projection $\mathbf{P}_r : \mathbb{R}^2 \to C(r)$ on the circle $C(r)$ is defined:*

$$\mathbf{P}_r(\boldsymbol{x}) = \begin{cases} \boldsymbol{x} & \text{for } ||\boldsymbol{x}|| \leq r, \\ \dfrac{r}{||\boldsymbol{x}||}\boldsymbol{x} & \text{for } ||\boldsymbol{x}|| > r. \end{cases}$$

Lemma 3 *The following statements are equivalent:*
a) The variational inequality from (3.5) is satisfied.
b) For all $i \in \mathcal{N}$ it holds:

$$||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| \leq g_i, \tag{3.6}$$

$$\mathbf{u}_{t,i} \neq \mathbf{0} \Rightarrow ||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| = g_i \quad \& \quad \mathbf{s}_{t,i} = k_i \mathbf{u}_{t,i}, \ k_i \geq 0. \tag{3.7}$$

c) For all $i \in \mathcal{N}$ it holds:

$$\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} = \mathbf{P}_{g_i}(\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \rho \mathbf{u}_{t,i}), \tag{3.8}$$

where $\rho > 0$ is any arbitrary but fixed parameter.

Proof. (i) We prove the implication "$a) \Rightarrow b)$" (the opposite implication is trivial). In the variational inequality (3.5) we choose the test vector as follows:

$$\mathbf{v} \in \mathbb{R}^{n_u} : \ \mathbf{v}_{t,j} = \mathbf{u}_{t,j}, \ j \neq i \ \text{ and } \ \mathbf{v}_{t,i} \in \mathbb{R}^2.$$

The variational inequality from (3.5) takes the form:

$$-(\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i})^{\mathsf{T}}(\mathbf{v}_{t,i} - \mathbf{u}_{t,i}) + g_i \left(||\mathbf{v}_{t,i}|| - ||\mathbf{u}_{t,i}||\right) \geq 0 \quad \forall \mathbf{v}_{t,i} \in \mathbb{R}^2. \tag{3.9}$$

If we choose $\mathbf{v}_{t,i} = \mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \mathbf{u}_{t,i}$ and use the triangular inequality, we can easily reach (3.6). We suppose that $\mathbf{u}_{t,i} \neq \mathbf{0}$. For $\mathbf{v}_{t,i} = \mathbf{0}$, we get from (3.9):

$$(\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i})^{\mathsf{T}} \mathbf{u}_{t,i} - g_i ||\mathbf{u}_{t,i}|| \geq 0 \tag{3.10}$$

and using the Cauchy inequality then

$$||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| - g_i \geq 0.$$

From here and (3.6) it follows $||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| = g_i$. If we use this result in (3.10) we get

$$(\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i})^{\mathsf{T}} \mathbf{u}_{t,i} - ||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| \cdot ||\mathbf{u}_{t,i}|| \geq 0. \tag{3.11}$$

Now we choose $\mathbf{v}_{t,i} = 2\mathbf{u}_{t,i}$ in (3.9) and we get:

$$-(\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i})^{\mathsf{T}} \mathbf{u}_{t,i} + ||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| \cdot ||\mathbf{u}_{t,i}|| \geq 0. \tag{3.12}$$

From (3.11) and (3.12) we get

$$(\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i})^{\mathsf{T}} \mathbf{u}_{t,i} = ||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| \cdot ||\mathbf{u}_{t,i}||.$$

Again, the Cachy inequality says that $\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}$ and $\mathbf{u}_{t,i}$ are the collinear vectors with the same orientation, and therefore $\mathbf{s}_{t,i} = k_i \mathbf{u}_{t,i},\ k_i \geq 0$.
(ii) We prove the implication "$c) \Rightarrow b)$" (the opposite implication is trivial). The inequality

$$||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| \leq g_i \tag{3.13}$$

follows direcly from (3.8). Let $\mathbf{u}_{t,i} \neq \mathbf{0}$. Then (3.5) yields $||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \rho \mathbf{u}_{t,i}|| \geq g_i$ and $||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| = g_i$. Using Definition 1 we can write

$$\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} = \frac{g_i}{||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \rho \mathbf{u}_{t,i}||} (\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \rho \mathbf{u}_{t,i})$$

and from here

$$(\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i})^{\mathsf{T}} (\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \rho \mathbf{u}_{t,i}) = ||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}|| \cdot ||\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \rho \mathbf{u}_{t,i}||.$$

And again, using the Cauchy inequality, we deduce that $\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i}$ and $\mathbf{s}_{t,i} - \kappa_i \mathbf{u}_{t,i} + \rho \mathbf{u}_{t,i}$ are consistently oriented collinear vectors and therefore $\mathbf{s}_{t,i} = k_i \mathbf{u}_{t,i},\ k_i \geq 0$. □

4 Algebraic problems and algorithms

4.1 Ossen iterations

Let us consider the problem:

$$
\begin{aligned}
&\textit{Find } (\mathbf{u},\mathbf{p},\mathbf{s}_n,\mathbf{s}_t) \in \mathbb{R}^{n_u}\times\mathbb{R}^{n_p}\times\mathbb{R}^{n_S}\times\mathbb{R}^{n_S} \textit{ such that}\\
&\mathbf{A}(\mathbf{u})\mathbf{u} + \mathbf{B}_u^{\mathsf{T}}(\mathbf{u})\mathbf{p} + \mathbf{N}^{\mathsf{T}}\mathbf{s}_n + \mathbf{T}^{\mathsf{T}}\mathbf{s}_t - \mathbf{b}(\mathbf{u}) = \mathbf{0},\\
&\mathbf{B}_l(\mathbf{u})\mathbf{u} - \mathbf{E}(\mathbf{u})\mathbf{p} - \mathbf{c}(\mathbf{u}) = \mathbf{0},\\
&\mathbf{N}\mathbf{u} = \mathbf{0},\\
&\left.\begin{aligned}
&(\mathbf{T}\mathbf{u})_i = 0 \Rightarrow |s_{ti}| \le g_i,\\
&(\mathbf{T}\mathbf{u})_i > 0 \Rightarrow s_{ti} = g_i + \kappa_i(\mathbf{T}\mathbf{u})_i,\\
&(\mathbf{T}\mathbf{u})_i < 0 \Rightarrow s_{ti} = -g_i + \kappa_i(\mathbf{T}\mathbf{u})_i,
\end{aligned}\right\} i \in \mathcal{N}.
\end{aligned}
$$

We will solve this problem using the Ossen iterations:

$$
\begin{aligned}
&\textit{Choose } \mathbf{u}^0 \in \mathbb{R}^{n_u}.\\
&\textit{Find } (\mathbf{u}^k,\mathbf{p}^k,\mathbf{s}_n^k,\mathbf{s}_t^k) \in \mathbb{R}^{n_u}\times\mathbb{R}^{n_p}\times\mathbb{R}^{n_S}\times\mathbb{R}^{n_S} \textit{ for } k \ge 1 \textit{ such that}\\
&\mathbf{A}^k\mathbf{u}^k + (\mathbf{B}_u^k)^{\mathsf{T}}\mathbf{p}^k + \mathbf{N}^{\mathsf{T}}\mathbf{s}_n^k + \mathbf{T}^{\mathsf{T}}\mathbf{s}_t^k - \mathbf{b}^k = \mathbf{0},\\
&\mathbf{B}_l^k\mathbf{u}^k - \mathbf{E}^k\mathbf{p}^k - \mathbf{c}^k = \mathbf{0},\\
&\mathbf{N}\mathbf{u}^k = \mathbf{0},\\
&\left.\begin{aligned}
&(\mathbf{T}\mathbf{u}^k)_i = 0 \Rightarrow |s_{ti}^k| \le g_i,\\
&(\mathbf{T}\mathbf{u}^k)_i > 0 \Rightarrow s_{ti}^k = g_i + \kappa_i(\mathbf{T}\mathbf{u}^k)_i,\\
&(\mathbf{T}\mathbf{u}^k)_i < 0 \Rightarrow s_{ti}^k = -g_i + \kappa_i(\mathbf{T}\mathbf{u}^k)_i,
\end{aligned}\right\} i \in \mathcal{N},
\end{aligned}
$$

where $\mathbf{A}^k = \mathbf{A}(\mathbf{u}^{k-1})$, $\mathbf{B}_u^k = \mathbf{B}_u(\mathbf{u}^{k-1})$, $\mathbf{B}_l^k = \mathbf{B}_l(\mathbf{u}^{k-1})$, $\mathbf{E}^k = \mathbf{E}(\mathbf{u}^{k-1})$, $\mathbf{b}^k = \mathbf{b}(\mathbf{u}^{k-1})$ and $\mathbf{c}^k = \mathbf{c}(\mathbf{u}^{k-1})$.

We rewrite the inner problem as a non-smooth equation, which we will solve by the semi-smooth Newton method. We define the function:

$$
\mathbf{G} : \mathbb{R}^N \to \mathbb{R}^N, \quad \mathbf{y} = (\mathbf{u}^{\mathsf{T}},\mathbf{p}^{\mathsf{T}},\mathbf{s}_n^{\mathsf{T}},\mathbf{s}_t^{\mathsf{T}})^{\mathsf{T}} \in \mathbb{R}^N
$$

given by

$$
\mathbf{G}^k(\mathbf{y}) = \begin{pmatrix}
\mathbf{A}^k\mathbf{u} + (\mathbf{B}_u^k)^{\mathsf{T}}\mathbf{p} + \mathbf{N}^{\mathsf{T}}\boldsymbol{\lambda}_n + \mathbf{T}^{\mathsf{T}}\mathbf{s}_t - \mathbf{b}^k\\
\mathbf{B}_l^k\mathbf{u} - \mathbf{E}^k\mathbf{p} - \mathbf{c}^k\\
\mathbf{N}\mathbf{u}\\
\boldsymbol{\Phi}(\mathbf{u},\mathbf{s}_t)
\end{pmatrix},
$$

where $N = n_u + n_p + 2n_S$. The function $\boldsymbol{\Phi}$, which ensures the satisfaction of the stick-slip

conditions, will be specified later. The Ossen iterations can be briefly written as follows:

$$\left.\begin{aligned} &\textit{Choose } \mathbf{u}^0 \in \mathbb{R}^{n_u}. \\ &\textit{Find } \mathbf{y}^k = \left((\mathbf{u}^k)^\mathsf{T}, (\mathbf{p}^k)^\mathsf{T}, (\mathbf{s}_n^k)^\mathsf{T}, (\mathbf{s}_t^k)^\mathsf{T}\right)^\mathsf{T} \textit{ for } k \geq 1 \textit{ such that } \mathbf{G}^k(\mathbf{y}^k) = \mathbf{0} \end{aligned}\right\} \tag{4.1}$$

4.2 Semi-smooth Newton method

Now we will explain the relations between the function $\mathbf{\Phi}$ and the stick-slip conditions, which we write for simplicity without the index k and for fixed $i \in \mathcal{N}$:

$$\left.\begin{aligned} &(\mathbf{Tu})_i = 0 \Rightarrow |s_{ti}| \leq g_i, \\ &(\mathbf{Tu})_i > 0 \Rightarrow s_{ti} = g_i + \kappa_i (\mathbf{Tu})_i, \\ &(\mathbf{Tu})_i < 0 \Rightarrow s_{ti} = -g_i + \kappa_i (\mathbf{Tu})_i. \end{aligned}\right\} \tag{4.2}$$

The relation between $(\mathbf{Tu})_i$ and s_{ti} expresses the blue graph in Figure 2, where the tangent of the line for $(\mathbf{Tu})_i < 0$ and $(\mathbf{Tu})_i > 0$ is κ_i.

Lemma 4 *The relation (4.2) is satisfied iff the pair $((\mathbf{Tu})_i, s_{ti})$ lies on the blue curve in Figure 2.*

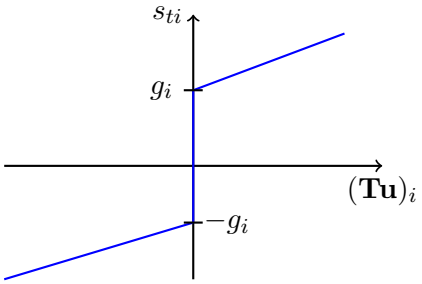


Figure 2: Relation between $(\mathbf{Tu})_i$ and s_{ti}.

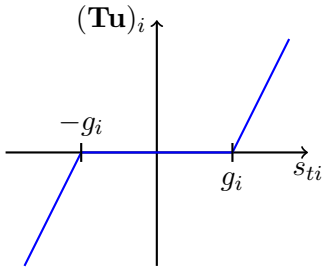


Figure 3: Relation between s_{ti} and $(\mathbf{Tu})_i$.

It is obvious that the relation between $(\mathbf{Tu})_i$ and s_{ti} is the multifunction, while the inverse relation, shown in Figure 3, is the function. The relation between s_{ti} and $(\mathbf{Tu})_i$ can be writen by parts:

$$(\mathbf{Tu})_i = \begin{cases} \kappa_i^{-1}(s_{ti} + g_i), & s_{ti} < -g_i, \\ 0, & s_{ti} \in \langle -g_i, g_i \rangle, \\ \kappa_i^{-1}(s_{ti} - g_i), & s_{ti} > g_i. \end{cases} \tag{4.3}$$

We can also write it by the max-function:

$$\phi : \mathbb{R} \to \mathbb{R}, \quad \phi(x) = \max\{0, x\}, \quad x \in \mathbb{R},$$

so that

$$(\mathbf{Tu})_i = \phi(\kappa_i^{-1}(s_{ti} - g_i)) - \phi(-\kappa_i^{-1}(s_{ti} + g_i)). \tag{4.4}$$

Lemma 5 *The pair* $(s_{ti}, (\mathbf{Tu})_i)$ *satisfies* (4.3) *iff it satisfies* (4.4).

Proof. If $s_{ti} < -g_i$, then $(\mathbf{Tu})_i = \kappa_i^{-1}(s_{ti} + g_i)$ according to (4.3). Since $s_{ti} - g_i < -2g_i \leq 0$, the first max-function in (4.4) is zero so that

$$(\mathbf{Tu})_i = -\phi(-\kappa_i^{-1}(s_{ti} + g_i)) = \kappa_i^{-1}(s_{ti} + g_i).$$

If $s_{ti} \in \langle -g_i, g_i \rangle$, then $(\mathbf{Tu})_i = 0$ according to (4.3). Since $s_{ti} - g_i \leq 0$ and $s_{ti} + g_i \geq 0$, both max-functions in (4.4) are zero. We get from (4.4):

$$(\mathbf{Tu})_i = 0.$$

If $s_{ti} > g_i$, then $(\mathbf{Tu})_i = \kappa_i^{-1}(s_{ti} - g_i)$ according to (4.3). Since $s_{ti} + g_i > 2g_i \geq 0$, the second max-function in (4.4) zero that

$$(\mathbf{Tu})_i = \kappa_i^{-1}(s_{ti} - g_i).$$

The lemma is proved. □

Lemma 6 *The relation* (4.2) *is satisfied iff* (4.4) *is satisfied.*

Proof. The pair $((\mathbf{Tu})_i, s_{ti})$ satisfies relation (4.2) iff the pair $(s_{ti}, (\mathbf{Tu})_i)$ satisfies the inverse relation. □

If we want to express the satisfied of relations (4.2) for all $i \in \mathcal{N}$, we use the vector notation.

Theorem 3 *The relations* (4.2) *are satisfied for all* $i \in \mathcal{N}$ *off*

$$\mathbf{\Phi}(\mathbf{u}, \mathbf{s}_t) = \mathbf{0},$$

where

$$\mathbf{\Phi}(\mathbf{u}, \mathbf{s}_t) = \mathbf{Tu} - \boldsymbol{\phi}(\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t - \mathbf{g})) + \boldsymbol{\phi}(-\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t + \mathbf{g}))$$

for $\mathbf{D}(\boldsymbol{\kappa}) = diag(\kappa_1, \ldots, \kappa_S) \in \mathbb{R}^{n_S \times n_S}$ *and* $\boldsymbol{\phi}(\mathbf{x}) = (\phi(x_1), \ldots, \phi(x_{n_S}))^{\mathsf{T}}$, $\mathbf{x} \in \mathbb{R}^{n_S}$.

The non-smooth equation from (4.1) will be solved by the Newton-type iterations. For the sake of simplicity we write this equation without the index k:

$$\mathbf{G}(\mathbf{y}) = \mathbf{0},$$

where function $\mathbf{G}$ has the block structure

$$\mathbf{G}(\mathbf{y}) = \left(\mathbf{G}_1(\mathbf{y})^{\mathsf{T}}, \mathbf{G}_2(\mathbf{y})^{\mathsf{T}}, \mathbf{G}_3(\mathbf{y})^{\mathsf{T}}, \mathbf{G}_4(\mathbf{y})^{\mathsf{T}}\right)^{\mathsf{T}},$$

and the blocks read as follows:

$$\begin{aligned}
\mathbf{G}_1(\mathbf{y}) &= \mathbf{A}\mathbf{u} + \mathbf{B}_u^\mathsf{T}\mathbf{p} + \mathbf{N}^\mathsf{T}\mathbf{s}_n + \mathbf{T}^\mathsf{T}\mathbf{s}_t - \mathbf{b},\\
\mathbf{G}_2(\mathbf{y}) &= \mathbf{B}_l\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c},\\
\mathbf{G}_3(\mathbf{y}) &= \mathbf{N}\mathbf{u},\\
\mathbf{G}_4(\mathbf{y}) &= \boldsymbol{\Phi}(\mathbf{u}, \mathbf{s}_t).
\end{aligned}$$

The Newton-type iterations generate the sequence $\{\mathbf{y}^l\},\ \mathbf{y}^l \in \mathbb{R}^N$, for the initial guess $\mathbf{y}^0 \in \mathbb{R}^N$ and consist in solving a sequence of linear systems:

$$\mathbf{G}^o(\mathbf{y}^l)\mathbf{y}^{l+1} = \mathbf{G}^o(\mathbf{y}^l)\mathbf{y}^l - \mathbf{G}(\mathbf{y}^l), \tag{4.5}$$

where $\mathbf{G}^o : \mathbb{R}^N \to \mathbb{R}^{N\times N}$ is the generalized Jacobian matrix (slanting function) to $\mathbf{G}$. The form of $\mathbf{G}^o$ will be got by differentiating $\mathbf{G}$. As the first three blocks in $\mathbf{G}$ are differentiable (linear), we use standard rules for their differentiating:

$$\begin{aligned}
\mathbf{G}_1^o(\mathbf{y}) &= (\mathbf{A}, \mathbf{B}_u^\mathsf{T}, \mathbf{T}^\mathsf{T}, \mathbf{N}^\mathsf{T}),\\
\mathbf{G}_2^o(\mathbf{y}) &= (\mathbf{B}_l, -\mathbf{E}, \mathbf{0}, \mathbf{0}),\\
\mathbf{G}_3^o(\mathbf{y}) &= (\mathbf{N}, \mathbf{0}, \mathbf{0}, \mathbf{0}).
\end{aligned}$$

For $\mathbf{G}_4$ we obtain $\mathbf{G}_4^o$ using the active/inactive sets and the respective indicator matrices. Remind that:

$$\mathbf{G}_4(\mathbf{y}) = \mathbf{T}\mathbf{u} - \boldsymbol{\phi}\left(\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t - \mathbf{g})\right) + \boldsymbol{\phi}\left(-\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t + \mathbf{g})\right).$$

Let $\mathcal{A}_t = \mathcal{A}_t(\mathbf{y})$, $\mathcal{I}_t^- = \mathcal{I}_t^-(\mathbf{y})$, and $\mathcal{I}_t^+ = \mathcal{I}_t^+(\mathbf{y})$ are the active and inactive sets at $\mathbf{y} \in \mathbb{R}^N$ defined by:

$$\begin{aligned}
\mathcal{A}_t &= \{i \in \mathcal{N} : s_{ti} \in \langle -g_i, g_i\rangle\},\\
\mathcal{I}_t^- &= \{i \in \mathcal{N} : s_{ti} < -g_i\},\\
\mathcal{I}_t^+ &= \{i \in \mathcal{N} : s_{ti} > g_i\},
\end{aligned}$$

where $\mathcal{N} = \{1, \ldots, n_S\}$. The indicator matrix for a subset $\mathcal{S} \subseteq \mathcal{N}$ is the diagonal matrix

$$\mathbf{D}(\mathcal{S}) = diag\,(s_1, \ldots, s_{n_S}),$$

where $s_i = 1$ for $i \in \mathcal{S}$ and $s_i = 0$ for $i \notin \mathcal{S}$.

From the active and inactive sets we get three indicator matrices $\mathbf{D}(\mathcal{A}_t)$, $\mathbf{D}(\mathcal{I}_t^+)$, and $\mathbf{D}(\mathcal{I}_t^-)$, by which we express $\mathbf{G}_4$ at $\mathbf{y} \in \mathbb{R}^N$ as follows:

$$\mathbf{G}_4(\mathbf{y}) = \mathbf{T}\mathbf{u} - \mathbf{D}(\mathcal{I}_t^+)\left(\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t - \mathbf{g})\right) + \mathbf{D}(\mathcal{I}_t^-)\left(-\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t + \mathbf{g})\right).$$

Now we can use again the standard differentiation rules:

$$\begin{aligned}
\frac{\partial \mathbf{G}_4(\mathbf{y})}{\partial \mathbf{u}} &= \mathbf{T}, \\
\frac{\partial \mathbf{G}_4(\mathbf{y})}{\partial \mathbf{p}} &= \mathbf{0}, \\
\frac{\partial \mathbf{G}_4(\mathbf{y})}{\partial \boldsymbol{\lambda}_n} &= \mathbf{0}, \\
\frac{\partial \mathbf{G}_4(\mathbf{y})}{\partial \mathbf{s}_t} &= -\mathbf{D}(\mathcal{I}_t^+)\mathbf{D}(\boldsymbol{\kappa})^{-1} - \mathbf{D}(\mathcal{I}_t^-)\mathbf{D}(\boldsymbol{\kappa})^{-1} = -\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-).
\end{aligned}$$

Summarizing these results we get

$$\mathbf{G}_4^o(\mathbf{y}) = \Big(\mathbf{T}, \mathbf{0}, \mathbf{0}, -\mathbf{D}(\boldsymbol{\kappa})\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-)\Big)$$

and the generalized Jacobian matrix in $\mathbf{y} \in \mathbb{R}^N$ read as follows:

$$\mathbf{G}_4^o(\mathbf{y}) = \left(\begin{array}{c|cc|c} \mathbf{A} & \mathbf{B}_u^\mathsf{T} & \mathbf{N}^\mathsf{T} & \mathbf{T}^\mathsf{T} \\ \hline \mathbf{B}_l & -\mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T} & \mathbf{0} & \mathbf{0} & -\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-) \end{array}\right).$$

For the right-hand side in (4.5) we calculate that

$$\mathbf{G}^o(\mathbf{y}^l)\mathbf{y}^l - \mathbf{G}(\mathbf{y}^l) = \Big(\mathbf{b}^\mathsf{T}, \mathbf{c}^\mathsf{T}, \mathbf{0}^\mathsf{T}, \Big(\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{D}(\mathcal{I}_t^-) - \mathbf{D}(\mathcal{I}_t^+))\mathbf{g}\Big)^\mathsf{T}\Big)^\mathsf{T}.$$

We have reached the followind „active/inactive set“ implementation of the semi-smooth Newton method (4.5), in which we use only the inactive sets.

Algorithm SSN (primar-dual version)

Choose $\mathbf{y}^0 = \Big((\mathbf{u}^0)^\mathsf{T}, (\mathbf{p}^0)^\mathsf{T}, (\mathbf{s}_n^0)^\mathsf{T}, (\mathbf{s}_t^0)^\mathsf{T}\Big)^\mathsf{T} \in \mathbb{R}^N$. For $l \geq 1$ compute:
(*Step* 1) Assembly the inactive sets at $\mathbf{y}^l = \Big((\mathbf{u}^l)^\mathsf{T}, (\mathbf{p}^l)^\mathsf{T}, (\boldsymbol{\lambda}_n^l)^\mathsf{T}, (\mathbf{s}_t^l)^\mathsf{T}\Big)^\mathsf{T}$:

$$\mathcal{I}_t^+ = \Big\{i \in \mathcal{N} : s_{ti}^l > g_i\Big\}, \quad \mathcal{I}_t^- = \Big\{i \in \mathcal{N} : s_{ti}^l < -g_i\Big\}$$

and the respective indicator matrices $\mathbf{D}(\mathcal{I}_t^+),\ \mathbf{D}(\mathcal{I}_t^-),\ \mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-)$.

(*Step* 2) Solve the linear system:

$$\left(\begin{array}{c|ccc} \mathbf{A} & \mathbf{B}_u^\mathsf{T} & \mathbf{N}^\mathsf{T} & \mathbf{T}^\mathsf{T} \\ \hline \mathbf{B}_l & -\mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T} & \mathbf{0} & \mathbf{0} & -\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-) \end{array}\right) \left(\begin{array}{c} \mathbf{u}^{l+1} \\ \mathbf{p}^{l+1} \\ \mathbf{s}_n^{l+1} \\ \mathbf{s}_t^{l+1} \end{array}\right) = \left(\begin{array}{c} \mathbf{b} \\ \mathbf{c} \\ \mathbf{0} \\ \mathbf{D}(\boldsymbol{\kappa})^{-1}\left(\mathbf{D}(\mathcal{I}_t^-) - \mathbf{D}(\mathcal{I}_t^+)\right)\mathbf{g} \end{array}\right).$$

4.3 Solving the linear systems

The efficiency of the SSN algorithm depends on the way how we solve the linear systems in *Step* 2. For simplicity, we introduce the matrices:

$$\mathbf{C}_u = \left(\begin{array}{c} \mathbf{B}_u \\ \mathbf{N} \\ \mathbf{T} \end{array}\right), \qquad \mathbf{C}_l = \left(\begin{array}{c} \mathbf{B}_l \\ \mathbf{N} \\ \mathbf{T} \end{array}\right), \qquad \bar{\mathbf{E}}^l = \left(\begin{array}{ccc} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-) \end{array}\right),$$

and the vectors

$$\mathbf{r}^{l+1} = \left(\begin{array}{c} \mathbf{p}^{l+1} \\ \mathbf{s}_n^{l+1} \\ \mathbf{s}_t^{l+1} \end{array}\right), \qquad \mathbf{h}^l = \left(\begin{array}{c} \mathbf{c} \\ \mathbf{0} \\ \mathbf{D}(\boldsymbol{\kappa})^{-1}\left(\mathbf{D}(\mathcal{I}_t^-) - \mathbf{D}(\mathcal{I}_t^+)\right)\mathbf{g} \end{array}\right).$$

Then the linear systems in *Step* 2 read as follows:

$$\left(\begin{array}{cc} \mathbf{A} & \mathbf{C}_u^\mathsf{T} \\ \mathbf{C}_l & -\bar{\mathbf{E}}^l \end{array}\right) \left(\begin{array}{c} \mathbf{u}^{l+1} \\ \mathbf{r}^{l+1} \end{array}\right) = \left(\begin{array}{c} \mathbf{b} \\ \mathbf{h}^l \end{array}\right).$$

Instead of this system we can solve its Shur complement:

$$\left(\mathbf{C}_l\mathbf{A}^{-1}\mathbf{C}_u^\mathsf{T} + \bar{\mathbf{E}}^l\right)\mathbf{r}^{l+1} = \mathbf{C}_l\mathbf{A}^{-1}\mathbf{b} - \mathbf{h}^l. \tag{4.6}$$

After solving (4.6), one can compute $\mathbf{u}^{l+1}$ using

$$\mathbf{u}^{l+1} = \mathbf{A}^{-1}\left(\mathbf{b} - \mathbf{C}_u^\mathsf{T}\mathbf{r}^{l+1}\right). \tag{4.7}$$

The Schur complement system (4.6) is non-symmetric, so that we will solve it by BiCGStab algorithm. For greater efficiency instead of demanding calculation of the inversion of the matrix $\mathbf{A}$, we will use the LU-decomposition of $\mathbf{A}$ and substitute $\mathbf{A}^{-1}$ as $\mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}$. Note that the matrices $\mathbf{A}$, $\mathbf{C}_u$, $\mathbf{C}_l$ and the vector $\mathbf{b}$ are not changed during the Newton iterations, therefore they are assembly only before the first Newton iteration. From (4.7) it is obvious that change of the vector $\mathbf{u}$ depend only on the change of the vector $\mathbf{r}$ therefore, it is enough to calculate it after the last *SSN* iteration.

After assembling all the matrices and the vectors we will start with SSN iterations, where as the first step is creating the inactive sets $\mathcal{I}_t^+$ and $\mathcal{I}_t^-$ and matrix $\mathbf{E}$.
The Schur complement $\mathbf{S} = \mathbf{C}_l\mathbf{A}^{-1}\mathbf{C}_u^\mathsf{T} + \bar{\mathbf{E}} = \mathbf{C}_l\mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}\mathbf{C}_u^\mathsf{T} + \bar{\mathbf{E}}$ is not assembled explicitely, but we will define its action on an arbitrary vector $\mathbf{x}$ as $\mathbf{S}(\mathbf{x}) = \mathbf{C}_l(\mathbf{U}\backslash\left(\mathbf{L}\backslash(\mathbf{P}(\mathbf{C}_u^\mathsf{T}\mathbf{x}))\right)) + \bar{\mathbf{E}}\mathbf{x}$, (here we have used the Backslash in the MATLAB solution for the backward substitutions). This calculation of the action of $\mathbf{S}$ is less demanding for memory and time. We safe the memory because we do not store the matrix $\mathbf{S}$ but only the vector $\mathbf{S}_\mathbf{x} = \mathbf{S}(\mathbf{x})$. Instead of calculating the inversions of the matrices $\mathbf{U}^{-1}$ and $\mathbf{L}^{-1}$, we solve two systems with the lower and upper triangular matrix $\mathbf{P}$ and $\mathbf{U}$ which we denoted by backslash.

4.4 Summary of algorithms

In this section we will present the implementation details of the three iterative algoritms that form the basis for solving our problem.

Algorithm Ossen

Input: $\mathbf{u}^0,\ \mathbf{r}^0,\ \mathbf{A}(\mathbf{u}),\ \mathbf{B}_u(\mathbf{u}),\ \mathbf{B}_l(\mathbf{u}),\ \mathbf{E}(\mathbf{u}),\ \mathbf{b}(\mathbf{u}),\ \mathbf{c}(\mathbf{u}),\ \mathbf{N},\ \mathbf{T},\ \mathbf{g},\ \boldsymbol{\kappa},\ tol,\ r_{tol},\ c_{fact} \in (0,1)$.
Output: $\mathbf{u},\ \mathbf{p},\ \mathbf{s}_n,\ \mathbf{s}_t$.
Set: $err^1 = 1,\ k = 1,\ ssntol^0 = r_{tol}/c_{fact}$ and compute:
(*Step* 1) If $err^k > tol$, then go to *Step* 2,
else return $\mathbf{u} = \mathbf{u}^{k-1},\ \mathbf{p} = \mathbf{p}^{k-1},\ \mathbf{s}_n = \mathbf{s}_n^{k-1},\ \mathbf{s}_t = \mathbf{s}_t^{k-1}$.
(*Step* 2) Assemble $\mathbf{A}^k = \mathbf{A}(\mathbf{u}^{k-1}),\ \mathbf{B}_u^k = \mathbf{B}_u(\mathbf{u}^{k-1}),\ \mathbf{B}_l^k = \mathbf{B}_l(\mathbf{u}^{k-1}),\ \mathbf{E}^k = \mathbf{E}(\mathbf{u}^{k-1})$,
$\mathbf{b}^k = \mathbf{b}(\mathbf{u}^{k-1}),\ \mathbf{c}^k = \mathbf{c}(\mathbf{u}^{k-1}),\ \mathbf{C}_u^k = \left((\mathbf{B}_u^k)^\mathsf{T}, \mathbf{N}^\mathsf{T}, \mathbf{T}^\mathsf{T}\right)^\mathsf{T},\ \mathbf{C}_l^k = \left((\mathbf{B}_l^k)^\mathsf{T}, \mathbf{N}^\mathsf{T}, \mathbf{T}^\mathsf{T}\right)^\mathsf{T}$.
Compute $\mathbf{P}^k,\ \mathbf{L}^k,\ \mathbf{U}^k$ which form LU-factorization of $\mathbf{A}^k$ such that $\mathbf{P}^k\mathbf{A}^k = \mathbf{L}^k\mathbf{U}^k$.
(*Step* 3) $ssntol^k = \min\left(r_{tol} \times err^k,\ c_{fact} \times ssntol^{k-1}\right)$.
(*Step* 4) Compute $\mathbf{h}^k = \mathbf{C}^k(\mathbf{A}^k)^{-1}\mathbf{b}^k - \left((\mathbf{c}^k)^\mathsf{T}, \mathbf{0}^\mathsf{T}, \mathbf{0}^\mathsf{T}\right)^\mathsf{T}$ and by the semi-smoth Newton method compute the second to fourth compoment $\mathbf{r}^k = \left((\mathbf{p}^k)^\mathsf{T}, (\mathbf{s}_n^k)^\mathsf{T}, (\mathbf{s}_t^k)^\mathsf{T}\right)^\mathsf{T}$ of the solution $\mathbf{y}^k = \left((\mathbf{u}^k)^\mathsf{T}, (\mathbf{r}^k)^\mathsf{T}\right)^\mathsf{T}$ to the equation $\mathbf{G}(\mathbf{y}^k) = \mathbf{0}$:

$$\mathbf{r}^k = \mathrm{SSN}\left(\mathbf{P}^k, \mathbf{L}^k, \mathbf{U}^k, \mathbf{C}_u^k, \mathbf{C}_l^k, \mathbf{h}^k, \mathbf{g}, \boldsymbol{\kappa},\ \mathbf{r}^0,\ ssntol^k\right).$$

(*Step* 5) Solve $\mathbf{u}^k = (\mathbf{A}^k)^{-1}\left(\mathbf{b}^k - (\mathbf{C}^k)^\mathsf{T}\mathbf{r}^k\right)$.
(*Step* 6) $err^{k+1} = ||\mathbf{y}^k - \mathbf{y}^{k-1}||/(||\mathbf{y}^k|| + 1)$, $k = k + 1$ and go to *Step* 1.

The algorithm of the semi-smooth Newton method is written here in the dual version, without generating the components $\mathbf{u}^l$.

Algorithm SSN(dual version)

Input: $\mathbf{P}, \mathbf{L}, \mathbf{U}, \mathbf{C}_u, \mathbf{C}_l, \mathbf{h}, \mathbf{g}, \boldsymbol{\kappa},\ \mathbf{r}^0,\ ssntol.$
Output: $\mathbf{r}$.
Set: $err^1 = 1,\ l = 1,\ bicgtol = r_{tol}/c_{fact}$ and compute:
(*Step* 1) If $err^l > ssntol$, then go to *Step* 2, else return $\mathbf{r}^{l-1}$.
(*Step* 2) Assemble the inactive sets $\mathcal{I}_t^+$ and $\mathcal{I}_t^-$ in r^{l-1} and then $\bar{\mathbf{E}}^l$.
(*Step* 3) Define function $\mathbf{S_x} = \mathbf{S}(\mathbf{x})$ and compute $\mathbf{d} = \mathbf{h} - \mathbf{h}^l$.
(*Step* 4) Solve the system of linear equations (4.6) with BiCGStab algorithm:

$$\mathbf{r}^l = \text{BiCGStab}\left(\mathbf{S_x}, \mathbf{d}, \mathbf{r}^{l-1}, bicgtol\right).$$

(*Step* 4) $err^{l+1} = ||\mathbf{r}^l - \mathbf{r}^{l-1}||/(||\mathbf{r}^l|| + 1),\ l = l + 1$ and go to *Step* 1.

Finaly we solve the linear system

$$\mathbf{Sr} = \mathbf{d},$$

by BiCGStab algorithm, where $\mathbf{S}$ is function of action of the Schur complement $\mathbf{S_x}$ on arbitrary vector $\mathbf{x}$. We will use the preconditioned BiCGStab algorithm with several different preconditioners, which we will specify later. In general, we describe the preconditioner $\mathbf{P}$ as well as the function of the Shur complement action $\mathbf{S_x} = \mathbf{S}(\mathbf{x})$ such that $\mathbf{P_x} = \mathbf{P}(\mathbf{x})$ and we denote $\mathbf{S}_\mathbf{x}^p = \mathbf{P}(\mathbf{S}(\mathbf{x}))$. The PBiCGStab algorithm is readed as follows:

Algorithm BiCGStab

Input: $\mathbf{S_x},\ \mathbf{d},\ \mathbf{r}^0,\ bicgtol$
Output: $\mathbf{r}$
Set: $err^1 = 1,\ k = 1,\ \mathbf{x} = \mathbf{r}^0,\ bicgtol = bicgtol\sqrt{\mathbf{d}^\mathsf{T}\mathbf{d}}$,
$\mathbf{r}_t = \mathbf{p} = \mathbf{r} = \mathbf{P}(\mathbf{d} - \mathbf{S}(\mathbf{x}))$.
(*Step* 1) If $err^k > bicgtol$, then go to *Step* 2, else return $\mathbf{r}$.
(*Step* 2) $\mathbf{S}_\mathbf{p}^p = \mathbf{P}(\mathbf{S}(\mathbf{p}))$,

$$\begin{aligned}
&r_{tr} = \mathbf{r}_t^\mathsf{T}\mathbf{r},\\
&\alpha = \frac{r_{tr}}{\mathbf{r}_t^\mathsf{T}\mathbf{S}_\mathbf{p}^p},\\
&\mathbf{s} = \mathbf{r} - \alpha\mathbf{S}_\mathbf{p}^p,\\
&\mathbf{S}_\mathbf{s}^p = \mathbf{P}(\mathbf{S}(\mathbf{s})),\\
&\omega = \frac{(\mathbf{S}_\mathbf{s}^p)^\mathsf{T}\mathbf{s}}{(\mathbf{S}_\mathbf{s}^p)^\mathsf{T}\mathbf{S}_\mathbf{s}^p},\\
&\mathbf{x} = \mathbf{x} + \alpha\mathbf{p} + \omega\mathbf{s},\\
&\mathbf{r} = \mathbf{s} - w\mathbf{S}_\mathbf{s}^p,\\
&\beta = \frac{\mathbf{r}_t^\mathsf{T}\mathbf{r}}{r_{tr}} \cdot \frac{\alpha}{\omega},\\
&\mathbf{p} = \mathbf{r} + \beta(\mathbf{p} - w\mathbf{S}_\mathbf{p}^p),
\end{aligned}$$

(*Step* 3) $err^{k+1} = ||\mathbf{r}||,\ k = k + 1$ and go to *Step* 1.

5 Numerical experiments in two dimensions

In this section we will perform several numerical experiments in two dimensions. First, we will test the convergence of the Ossen iterations for the Navier-Stokes problem. Then we will compare the number of the Ossen iterations with different preconditions of internal solvers for the case with and without the stick-slip condition on square, rectangular and L-shape domain Ω.

5.1 Convergence of the Ossen iterations

Let's have the nonlinear Navier-Stokes equation as follows:

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p &= \boldsymbol{f} && \text{in } \Omega, \\ \nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{u}_D && \text{on } \partial\Omega, \end{aligned} \tag{5.1}$$

where we consider the domain $\Omega = (0,1)^2$ with the Dirichlet boundary condition $\mathbf{u}_D = 0$ on $\partial\Omega$, $\boldsymbol{u} : \overline{\Omega} \to \mathbb{R}^2$, $p : \overline{\Omega} \to \mathbb{R}$, and viscosity ν with values $1, 0.1, 0.01$ and 0.001. The right-hand side $\boldsymbol{f}$ is adjusted such that analytical solution as follow:

$$\begin{aligned} u_{exp_1}(x_1,x_2) &= -\cos(2\pi x_1)\sin(2\pi x_2) + \sin(2\pi x_2), \\ u_{exp_2}(x_1,x_2) &= \sin(2\pi x_1)\cos(2\pi x_2) - \sin(2\pi x_1), \\ p_{exp}\ (x_1,x_2) &= 2\pi\left(\cos(2\pi x_2) - \cos(2\pi x_1)\right), \end{aligned}$$

where u_{exp_1}, u_{exp_2} represent the velocity in x_1, x_2 direction, respectively. The graphical representation of the velocity field is shown in Figure 4.

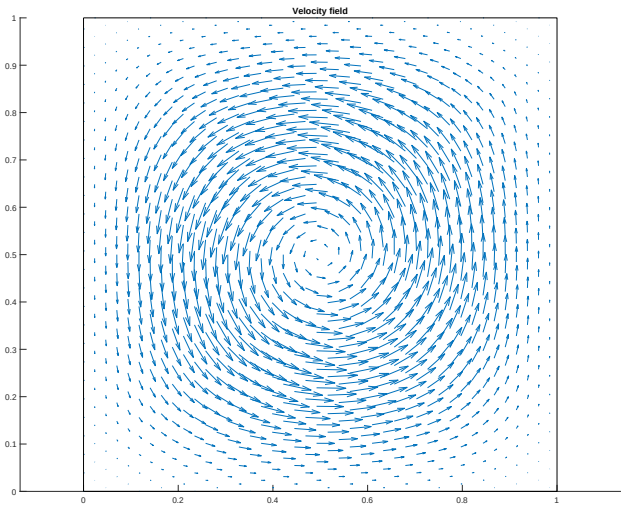


Figure 4: Velocity field $\nu = 1$

We will linearize the convective term $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$ in (5.1) such that we get the Ossen problem, where the divergence free function $\boldsymbol{w}$ is given:

$$\begin{aligned} -\nu \Delta \boldsymbol{u} + \boldsymbol{w} \cdot \nabla \boldsymbol{u} + \nabla p &= \boldsymbol{f} \qquad \text{in } \Omega, \\ \nabla \cdot \boldsymbol{u} &= 0 \qquad \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{u}_D \qquad \text{on } \partial\Omega \end{aligned} \tag{5.2}$$

and we approximate it by the FEM with the bubble functions. We get the linear system writen as follows

$$\begin{aligned} \mathbf{A}(\mathbf{w})\mathbf{u} + \mathbf{B}_u(\mathbf{w})^{\mathsf{T}}\mathbf{p} &= \mathbf{b}(\mathbf{w}), \\ \mathbf{B}_l(\mathbf{w})\mathbf{u} - \mathbf{E}(\mathbf{w})\mathbf{p} &= \mathbf{c}(\mathbf{w}). \end{aligned} \tag{5.3}$$

We will solve this linear system using the simpler version of the Ossen iterations, where the main problem in *Step* 2 is solved by direct solver (DIR), which is represented in the MATLAB by the Backslash. Note that the norm in *Step* 3 is Euclidean.

Ossen iterations (Simpler version)

Input: $\mathbf{u}^1$.
Set: $\mathbf{w} = \mathbf{0}$, $err^1 = 1$, $k = 1$ and compute.
(*Step* 1) If $err^k > 10^{-4}$, then go to *Step* 2, else return $\mathbf{u} = \mathbf{u}^k, \mathbf{p} = \mathbf{p}^k$.
(*Step* 2) Solve the system of linear equation (5.3) with direct solver (DIR):

$$\begin{pmatrix} \mathbf{A}(\mathbf{u}^k) & \mathbf{B}_u(\mathbf{u}^k)^{\mathsf{T}} \\ \mathbf{B}_l(\mathbf{u}^k) & -\mathbf{E}(\mathbf{u}^k) \end{pmatrix} \begin{pmatrix} \mathbf{u}^{k+1} \\ \mathbf{p}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}(\mathbf{u}^k) \\ \mathbf{c}(\mathbf{u}^k) \end{pmatrix}.$$

(*Step* 3) Compute err^{k+1}:

$$err^{k+1} = \frac{||(\mathbf{u}^{k+1}, \mathbf{p}^{k+1}) - (\mathbf{u}^k, \mathbf{p}^k)||}{||(\mathbf{u}^{k+1}, \mathbf{p}^{k+1})||}.$$

(*Step* 4) $k := k + 1$ and go to *Step* 1.

We will test the order of convergence for the two norms e_1 and e_2 as follows:

$$\begin{aligned} e_1(h) &= ||\boldsymbol{u}_h - \boldsymbol{u}_{exp}||_{(L^2(\Omega))^2}, \\ e_2(h) &= ||p_h - p_{exp}||_{L^2(\Omega)} + ||\boldsymbol{u}_h - \boldsymbol{u}_{exp}||_{(H^1(\Omega))^2}. \end{aligned}$$

The order of the convergence $p \in \mathbb{R}$ is defined as follows:

$$e(h) \leq Ch^p, \tag{5.4}$$

where $h \in \mathbb{R}$ is the parameter of discretization, $C \in \mathbb{R}$ is the constant independent of h.

If we choose the worst estimate then (5.4) changes to the equation. Let us choose $h_j = \frac{1}{2}h_{j-1}$. For h_j and h_{j-1} we get two equations from (5.4) which we divide and substitute readed as follows:

$$\frac{e(h_{j-1})}{e(h_j)} \doteq \frac{Ch_{j-1}^p}{Ch_j^p} = \frac{C(2h_j)^p}{Ch_j^p} = 2^p. \tag{5.5}$$

From exponencial equation (5.5) and for $e(h) = e_1(h)$ we obtain:

$$p_{1,j} := \log_2 \left(\frac{e_1(h_{j-1})}{e_1(h_j)} \right) \tag{5.6}$$

and similary we get $p_{2,j}$ for $e(h) = e_2(h)$.

The order of convergences $p_{1,j}$ and $p_{2,j}$ were calculated for different discretizations with parameter h_j, $j = 1, \ldots, 8$, which is size of the largest edge of triangular elements of the discretization. The results are in Tables 1, 2 and graphical comparsion are in Figures 5-8.

Table 1: p_1 and p_2 for $\nu = 1,\ 0.1$.

ν	1				0.1			
h_j	e_1	p_1	e_2	p_2	e_1	p_1	e_2	p_2
1/4	0.36297	-	7.8298	-	0.35978	-	6.3911	-
1/8	0.10687	1.7640	4.2243	0.8903	0.10705	1.7488	3.5154	0.8624
1/16	0.02781	1.9424	1.9838	1.0905	0.02777	1.9465	1.7320	1.0213
1/32	0.00702	1.9869	0.9356	1.0843	0.00699	1.9895	0.8458	1.0340
1/64	0.00175	1.9990	0.4495	1.0577	0.00175	1.9999	0.4162	1.0229
1/128	0.00044	2.0011	0.2186	1.0401	0.00044	2.0014	0.2062	1.0131
1/256	0.00011	2.0009	0.1072	1.0281	0.00011	2.0011	0.1026	1.0071
1/512	0.00003	2.0006	0.0529	1.0198	0.00003	2.0006	0.0512	1.0039

Table 2: p_1 and p_2 for $\nu = 0.01,\ 0.001$.

ν	0.01				0.001			
h_j	e_1	p_1	e_2	p_2	e_1	p_1	e_2	p_2
1/4	0.34642	-	6.5078	-	1.14502	-	20.7754	-
1/8	0.11497	1.5913	3.7909	0.7796	0.02538	2.1738	8.1526	1.3495
1/16	0.032490	1.8231	2.0119	0.9140	0.08733	1.5390	5.3553	0.6063
1/32	0.00727	2.1607	0.8748	1.2015	0.03706	1.2367	4.4261	0.2749
1/64	0.00173	2.0669	0.4115	1.0882	0.00463	3.0005	1.0948	2.0154
1/128	0.00043	2.0198	0.2014	1.0306	0.00062	2.9176	0.2883	1.9251
1/256	0.00011	2.0049	0.0999	1.0103	0.00012	2.3766	0.1107	1.3809
1/512	0.00003	1.9978	0.0499	1.0005	0.00003	2.1270	0.0506	1.1294

The results show that for any ν with corresponding approximation the following relations

apply:

$$\lim_{h_j \to 0} p_1 = 2,$$

$$\lim_{h_j \to 0} p_2 = 1,$$

which is consistent with experimental observations on superconvergence of FEM for MINI elements (P1-bubble/P1) for the Stokes problem [9]. In ourcase we observe superconvergence result for the Navier-Stokes problem.

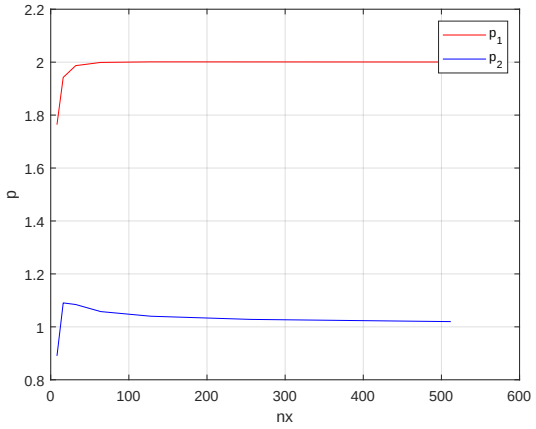


Figure 5: p_1 and p_2 for $\nu = 1$.

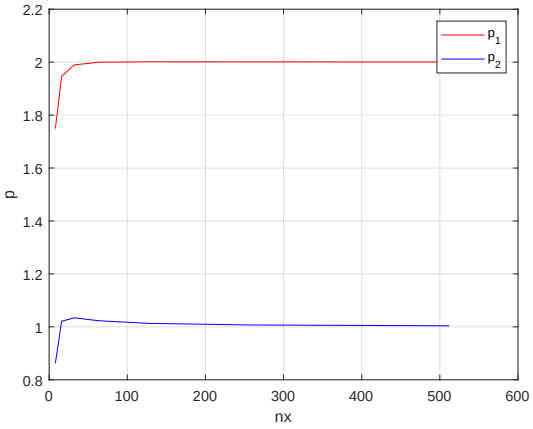


Figure 6: p_1 and p_2 for $\nu = 0.1$.

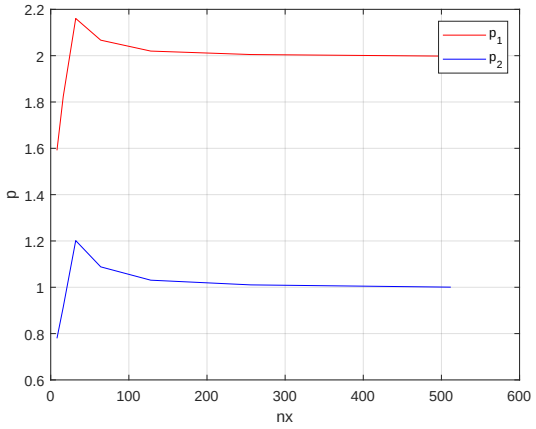


Figure 7: p_1 and p_2 for $\nu = 0.01$.

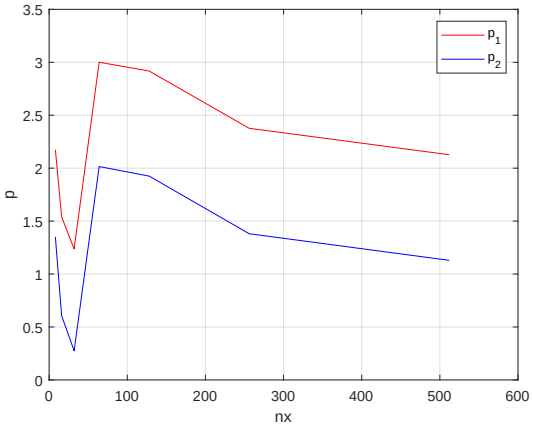


Figure 8: p_1 and p_2 for $\nu = 0.001$.

5.2 Dirichlet-Neumann boundary conditions

Let's have the nonlinear Navier-Stokes equation on the bounded domain $\Omega \subset \mathbb{R}^2$, with boundary $\partial\Omega$, that is split into two disjoint parts γ_D, and γ_N, such that $\partial\Omega = \overline{\gamma_D} \cup \overline{\gamma_N}$, write as follow:

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p &= \boldsymbol{f} && \text{in } \Omega, \\ \nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{u}_D && \text{on } \gamma_D, \\ \boldsymbol{\sigma} &= \boldsymbol{\sigma}_N && \text{on } \gamma_N, \end{aligned} \tag{5.7}$$

where the right-hand side $\boldsymbol{f} = \boldsymbol{0}$, $\boldsymbol{u} : \overline{\Omega} \to \mathbb{R}^2$, $p : \overline{\Omega} \to \mathbb{R}$ and $\nu > 0$. The Neuman boundary condition is equal $\boldsymbol{\sigma}_N = \boldsymbol{0}$ on γ_N. Dirichlet boundary has two disjoint parts as well, $\gamma_D = \overline{\gamma_{D_{in}}} \cup \overline{\gamma_{D_{wall}}}$. On $\gamma_{D_{wall}}$ is $\boldsymbol{u}_D = \boldsymbol{0}$, and on $\gamma_{D_{in}}$ is $\boldsymbol{u}_D$ described by a parabolic function which is defined as follows:

$$\boldsymbol{u}_{D_{in}}(x_1, x_2) = u_{in}(x_2 - x_{2_{min}})(x_2 - x_{2_{max}}), \tag{5.8}$$

where $u_{in} \in \mathbb{R}$ is the constant that describes the magnitude of the input speed, and $x_{2_{min}}$, $x_{2_{max}}$ are the minimum, maximum x_2 coordinate of the $\gamma_{D_{in}}$, respectively.

We linearize the problem (5.7) in the same way as the problem (5.1) in Section 5.1 and we get:

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + \boldsymbol{w}\cdot\nabla\boldsymbol{u} + \nabla p &= \boldsymbol{f} && \text{in } \Omega, \\ \nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{u}_D && \text{on } \gamma_D, \\ \boldsymbol{\sigma} &= \boldsymbol{\sigma}_N && \text{on } \gamma_N, \end{aligned} \tag{5.9}$$

where $\boldsymbol{w}$ is given. Then, with the finite element method, we obtain a formally identical system as (5.3). Finally, we solve this system by the simpler version of the Ossen iteration, which are described in detail in the Section 5.1. Next, we will also use the BiCGstab algorithm, which is described in the Section 4.4, instead of the direct solver to solve the system in the *Step* 2.

As in Section 4.3, instead of this system:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_u^\mathsf{T} \\ \mathbf{B}_l & -\mathbf{E} \end{pmatrix} \begin{pmatrix} \mathbf{u}^{k+1} \\ \mathbf{p}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix},$$

we can solve its Shur complement system:

$$(\mathbf{B}_l\mathbf{A}^{-1}\mathbf{B}_u^\mathsf{T} + \mathbf{E})\mathbf{p}^{k+1} = \mathbf{B}_l\mathbf{A}^{-1}\mathbf{b} - \mathbf{c}, \tag{5.10}$$

by BiCGstab algorithm and then compute:

$$\mathbf{u}^{k+1} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{B}_u^{\mathsf{T}}\mathbf{p}). \tag{5.11}$$

Similarly, we use the substitution via the LU-decomposition of $\mathbf{A}$, instead of $\mathbf{A}^{-1}$ and we work with the Shur complement action on an arbitrary vector $\mathbf{x}$ only. The BiCGstab ending-tolerance is computed and modified in each Ossen iteration. The modified simpler Ossen algorithm is write as follow:

Ossen iterations with BiCGstab

Input: $\mathbf{u}^0,\ \mathbf{p}^0,\ tol,\ r_{tol},\ c_{fact} \in (0,1)$.
Output: $\mathbf{u},\ \mathbf{p}$.
Set: $err^1 = 1,\ k = 1,\ bicgtol^0 = r_{tol}/c_{fact},\ \mathbf{y}^0 = \left((\mathbf{u}^0)^{\mathsf{T}}, (\mathbf{p}^0)^{\mathsf{T}}\right)^{\mathsf{T}}$ and compute:
(*Step* 1) If $err^k > tol$, then go to *Step* 2, else return $\mathbf{u} = \mathbf{u}^k,\ \mathbf{p} = \mathbf{p}^k$.
(*Step* 2) Assemble $\mathbf{A}^k = \mathbf{A}(\mathbf{u}^k),\ \mathbf{B}_l^k = \mathbf{B}_l(\mathbf{u}^k),\ \mathbf{B}_u^k = \mathbf{B}_u(\mathbf{u}^k),\ \mathbf{E}^k = \mathbf{E}(\mathbf{u}^k)$, $\mathbf{b}^k = \mathbf{b}(\mathbf{u}^k),\ \mathbf{c}^k = \mathbf{c}(\mathbf{u}^k)$, and compute $\mathbf{P}^k,\ \mathbf{L}^k,\ \mathbf{U}^k$ which form LU-factorization of $\mathbf{A}^k$ such that $\mathbf{P}^k\mathbf{A}^k = \mathbf{L}^k\mathbf{U}^k$.
(*Step* 3) $bicgtol^k = \min\left(r_{tol} \times err^k,\ c_{fact} \times bicgtol^{k-1}\right)$.
(*Step* 4) Compute $\mathbf{p}^{k+1}$ by the BiCGstab algorithm:

$$\mathbf{p}^{k+1} = \text{BiCGstab}\left(\mathbf{P}^k, \mathbf{L}^k, \mathbf{U}^k, \mathbf{B}_l^k, \mathbf{B}_u^k, \mathbf{b}^k, \mathbf{c}^k,\ \mathbf{r}^0,\ bicgtol^k\right).$$

(*Step* 5) Solve $\mathbf{u}^k = (\mathbf{A}^k)^{-1}\left(\mathbf{b}^k - (\mathbf{B}_u^k)^{\mathsf{T}}\mathbf{p}^k\right),\ \mathbf{y}^k = \left((\mathbf{u}^k)^{\mathsf{T}}, (\mathbf{p}^k)^{\mathsf{T}}\right)^{\mathsf{T}}$.
(*Step* 6) $err^{k+1} = ||\mathbf{y}^k - \mathbf{y}^{k-1}||/(||\mathbf{y}^k|| + 1)$, $k = k + 1$ and go to *Step* 1.

We will use the pure BiCGstab and also preconditioned BICGstab with two type of preconditioners. So we will test four options here in the *Step* 4:

- Direct solver (Backslash in the MATLAB)
- Pure BICGstab (without): $\mathbf{P}_x = \mathbf{x}$
- Mass matrix: $\mathbf{P}_x = \text{diag}(\mathbf{M}).\backslash\mathbf{x}$.
- Lumped: $\mathbf{P}_x = \mathbf{B}_l\mathbf{A}\left(\mathbf{x}^{\mathsf{T}}\mathbf{B}_u\right)^{\mathsf{T}} + \text{diag}(\mathbf{E}).\backslash\mathbf{x}$

We will compare the number of the Ossen iterations it, the number of matrix multiplications n_{MV} and also the time requirement on the rectangle and L-shape as follows.

Rectangular domain

Let's consider the rectangular domain $\Omega = (0,5) \times (0,1)$ which is in the Figure 9. The triangular mesh is uniform in this case and one is shown in the Figure 10.

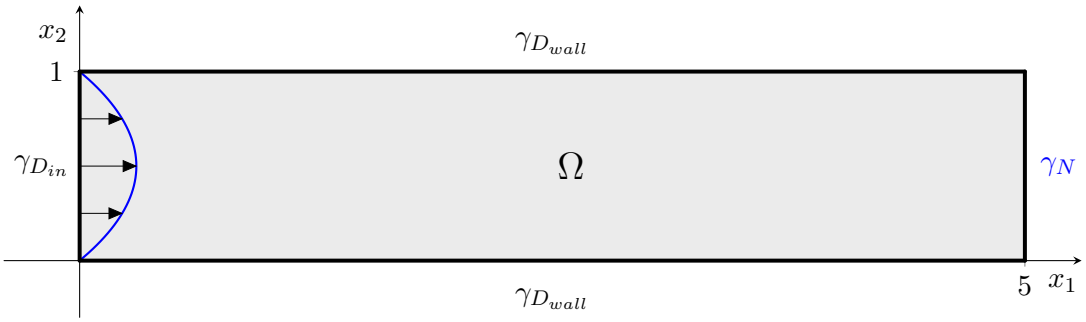


Figure 9: Rectangular domain with DN conditions

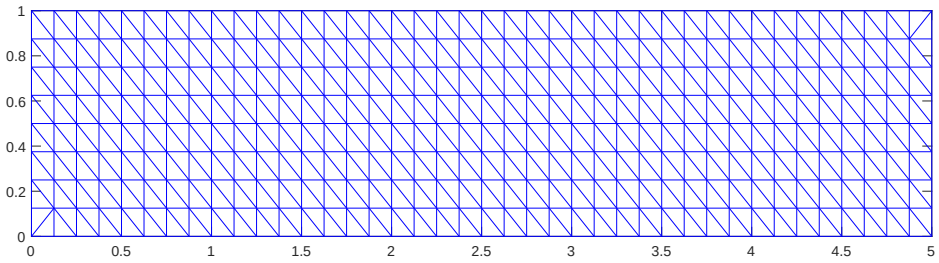


Figure 10: Mesh of the rectangular domain

For this domain $x_{2_{min}} = 0$ and $x_{2_{max}} = 1$. We choose $u_{in} = 0.4$, then the inlet parabolic function (5.8) will be in the form as follows:

$$\boldsymbol{u}_{D_{in}}(x_1, x_2) = 0.4x_2(x_2 - 1).$$

The experiments runs with precision $\epsilon = 10^{-4}$ and ν with values $1, 0.1, 0.01$ and 0.001 for the direct solver and BICGstab solver with preconditioner prescribed above.

The results are shown in tables 3-6, where n_p, n_t denotes the number of nodes, triangles, respectively, and it,n_{MV} number of the Ossen iterations and the multiplication of matrices respectively.

We also calculate the Reynolds number $Re = \frac{uL}{\nu}$, where L is a characteristic linear dimension. The Reynolds number is a dimensionless quantity that relates the inertial forces and the viscosity (i.e. the resistance of the environment due to internal friction) and helps predict flow patterns in different fluid flow situations. With its increasing value, the flow ceases to be laminar and our models cease to converge. It can be used to determine whether the fluid flow is laminar or turbulent.

If we ignore the direct solver, then the result shown that the preconditioned BiCGstab is faster then clear BiCGstab. For $\nu = 1, 0.1$ is the mass matrix and lumped preconditioners have similar results and for $\nu < 0.1$ is mass matrix preconditioner better.

The calculated velocity field with velocity arrows is shown in Figure 11.

Table 3: DN Rectangular domain $\nu = 1$, $Re = 0.05$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
33/40	2/0	0.02	7/83	0.02	4/76	0.02	3/71	0.02
105/160	2/0	0.03	7/101	0.03	3/81	0.03	3/85	0.03
369/640	2/0	0.06	7/109	0.13	3/99	0.08	2/64	0.03
1377/2560	2/0	0.16	7/99	0.78	3/113	0.48	2/64	0.28
5313/10240	2/0	0.86	8/132	8.42	2/68	2.67	2/76	2.59
20865/40960	2/0	3.98	8/122	113.24	2/84	33.14	2/86	32.34

Table 4: DN Rectangular domain $\nu = 0.1$, $Re = 0.5$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
33/40	3/0	0.01	5/73	0.03	4/82	0.00	4/80	0.01
105/160	3/0	0.02	6/112	0.02	3/79	0.01	4/100	0.01
369/640	2/0	0.05	6/106	0.09	3/95	0.06	3/89	0.08
1377/2560	2/0	0.16	7/155	0.87	3/109	0.48	3/101	0.44
5313/10240	2/0	0.73	7/127	7.50	2/64	2.61	2/60	2.42
20865/40960	2/0	3.95	7/135	99.95	2/84	33.15	2/74	31.31

Table 5: DN Rectangular domain $\nu = 0.01$, $Re = 5$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
33/40	3/0	0.02	5/95	0.00	3/67	0.02	5/83	0.00
105/160	3/0	0.02	5/125	0.03	3/109	0.21	4/104	0.02
369/640	2/0	0.06	6/170	0.13	3/153	0.08	4/146	0.08
1377/2560	2/0	0.16	6/170	0.79	3/171	0.58	3/115	0.47
5313/10240	2/0	0.75	7/199	8.25	2/94	2.93	3/143	4.24
20865/40960	2/0	3.96	7/147	101.06	2/118	35.65	3/159	50.97

Table 6: DN Rectangular domain $\nu = 0.001$, $Re = 50$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
33/40	3/0	0.02	5/103	0.00	4/144	0.02	8/162	0.03
105/160	4/0	0.02	7/417	0.05	4/374	0.05	8/478	0.06
369/640	3/0	0.06	6/578	0.19	3/547	0.13	6/692	0.19
1377/2560	2/0	0.16	7/1525	2.92	3/1125	2.01	5/1197	2.39
5313/10240	2/0	0.73	6/1100	17.03	2/616	8.55	4/1386	19.31
20865/40960	2/0	3.95	7/1199	181.40	2/640	76.53	3/1109	126.59

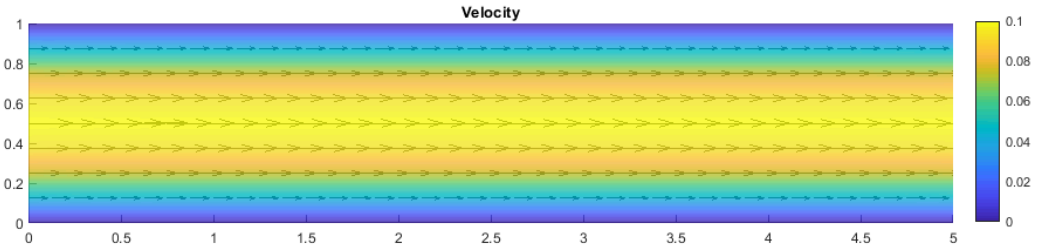


Figure 11: The velocity field for $\nu = 0.01$ and $u_{max} = 0.1$, $Re = 5$

L-shaped domain

Next, we will have L-shaped domain $\Omega = (0,5) \times (0,2) \backslash (0,1) \times (0,1)$ which is in Figure 12, and one of the triangular meshes is shown in Figure 13.

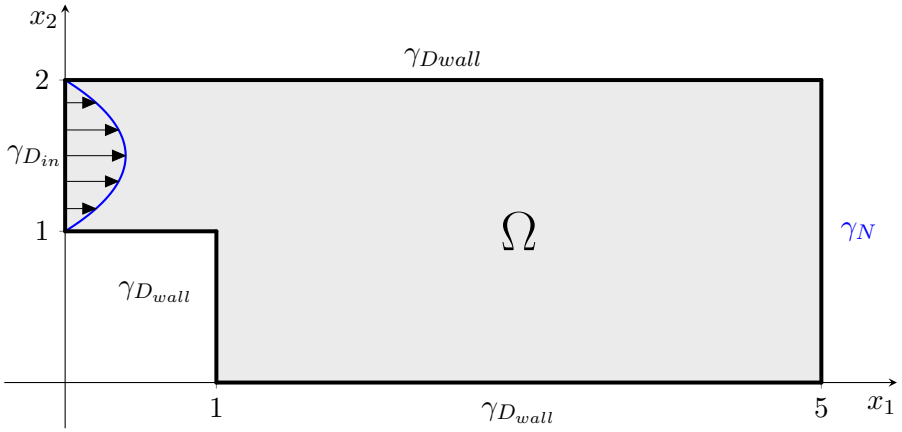


Figure 12: L-shaped domain with DN conditions

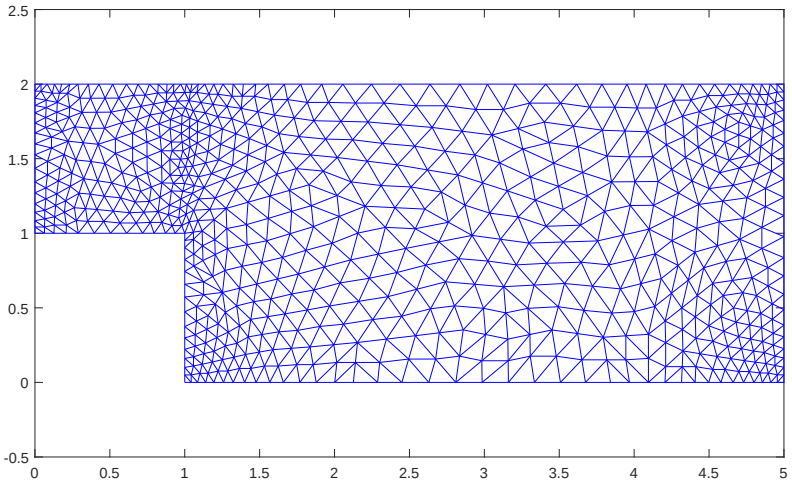


Figure 13: Mesh the L-shaped domain

We will perform the same experiments as on the rectangle, with same values of the constants ν. The inlet parabolic function will have form as follows:

$$\boldsymbol{u}_{D_{in}}(x_1, x_2) = 0.4(x_2 - 1)(x_2 - 2).$$

The complete results are shown in Tables 7-10 and the calculated velocity field is shown in Figure 14. It can be seen that the lumped preconditioner is best opinion for L-shaped Ω.

Table 7: DN L-shaped $\nu = 1$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
174/278	3/0	0.03	8/320	0.10	3/69	0.03	3/99	0.03
206/333	3/0	0.04	8/350	0.10	3/69	0.04	3/101	0.03
744/1332	3/0	0.14	8/428	0.31	3/89	0.13	3/119	0.12
2819/5328	3/0	0.68	8/412	2.04	3/115	0.81	3/139	0.78
10965/21312	3/0	3.51	8/454	14.09	3/127	5.40	3/147	5.23
43241/85248	3/0	18.82	8/454	105.85	3/127	39.34	3/149	38.54

Table 8: DN L-shaped $\nu = 0.1$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
174/278	3/0	0.03	7/343	0.07	4/108	0.03	4/104	0.03
206/333	3/0	0.04	7/369	0.09	4/108	0.05	4/104	0.04
744/1332	3/0	0.14	7/333	0.27	3/87	0.13	4/128	0.16
2819/5328	3/0	0.68	8/496	2.19	3/95	0.78	3/95	0.68
10965/21312	3/0	3.52	8/412	13.54	3/103	5.10	3/103	4.60
43241/85248	3/0	18.79	8/442	104.71	3/111	37.76	3/109	35.46

Table 9: DN L-shaped $\nu = 0.01$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
174/278	5/0	0.04	6/232	0.06	5/163	0.04	5/103	0.04
206/333	5/0	0.06	7/349	0.09	5/165	0.07	5/113	0.05
744/1332	5/0	0.24	7/357	0.27	5/229	0.21	5/165	0.19
2819/5328	5/0	1.16	7/391	1.84	5/249	1.38	5/193	1.22
10965/21312	6/0	7.04	8/478	14.39	6/390	11.88	6/296	10.65
43241/85248	6/0	37.54	8/476	107.70	6/394	86.75	6/338	81.37

Table 10: DN L-shaped $\nu = 0.001$

Solver	DIR		BiCGstab		Mass matrix		Lumped	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
174/278	11/0	0.10	11/801	0.13	11/1087	0.13	17/805	0.17
206/333	11/0	0.13	14/1216	0.21	11/1151	0.16	16/912	0.21
744/1332	16/0	0.74	16/2894	0.93	16/3644	1.08	12/1178	0.68
2819/5328	17/0	3.97	15/3683	8.96	17/6369	14.40	17/4215	11.66
10965/21312	18/0	21.15	15/4241	69.60	18/9276	137.18	18/6678	113.86
43241/85248	18/0	112.62	14/3628	422.25	18/10414	1035.59	18/7852	858.44

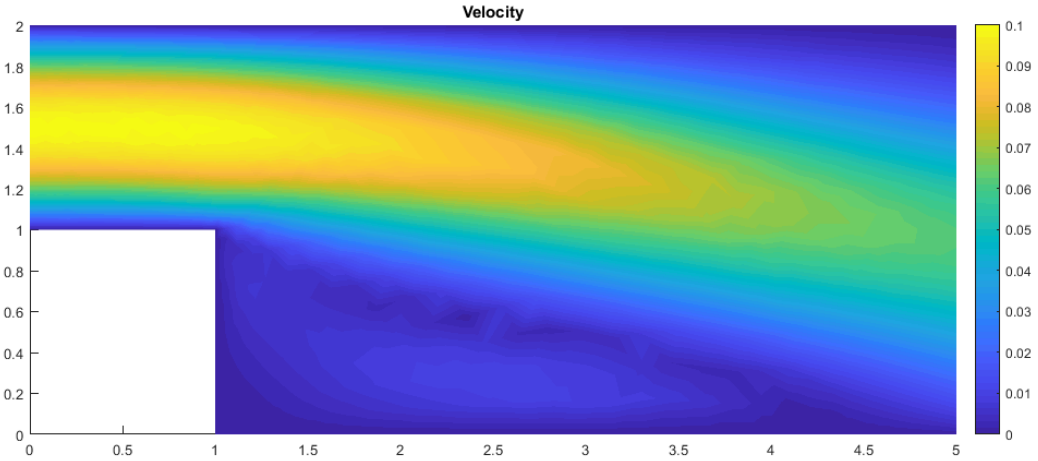


Figure 14: The velocity field for $\nu = 0.001$ and $u_{in} = 0.4$

5.3 DN and stick-slip boundary conditions

In this section we will do experiments with the nonlinear Navier-Stokes equation on the bounded domain $\Omega \subset \mathbb{R}^2$ with Dirichlet, Neumann and stick-slip boundary conditions (5.12). Detailed description of (5.12) is described in Sections 2-4, i.e. weak formulation, discretization, the Ossen iterations, ect. Let us remind only its wording, write as follows:

$$\begin{aligned}
-\nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p &= \boldsymbol{f} && \text{in } \Omega,\\
\nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega,\\
\boldsymbol{u} &= \boldsymbol{u}_D && \text{on } \gamma_D,\\
\boldsymbol{\sigma} &= \boldsymbol{\sigma}_N && \text{on } \gamma_N,\\
u_n &= 0 && \text{on } \gamma_S,\\
||\boldsymbol{\sigma}_t + \kappa\boldsymbol{u}_t|| &\le g && \text{on } \gamma_S,\\
\boldsymbol{\sigma}_t\cdot\boldsymbol{u}_t + g||\boldsymbol{u}_t|| + \kappa\boldsymbol{u}_t\cdot\boldsymbol{u}_t &= 0 && \text{on } \gamma_S.
\end{aligned} \tag{5.12}$$

The boundary of Ω is split into three disjoint parts γ_D, γ_N, and γ_S, such that $\partial\Omega = \overline{\gamma_D} \cup \overline{\gamma_N} \cup \overline{\gamma_S}$. Similar to the previous examples is $\gamma_D = \overline{\gamma_{D_{in}}} \cup \overline{\gamma_{D_{wall}}}$.

Let's choose $\mathbf{u}_D = \mathbf{0}$ on $\gamma_{D_{wall}}$ and $\boldsymbol{u}_D$ is the same described by a parabolic function on $\gamma_{D_{in}}$ as in the Section 5.2.

The outer solver will be semi-smooth Newton method and the inner solver BICGstab, which we will test for differend preconditioners, as follow:

- Pure BICGstab (without)
- Mass matrix
- Lumped
- Diagonal

Preliminary tests revealed that the mass preconditioner diverges or calculates bad results for any setting, so it was excluded from the main tests.

We will work with squared, rectangle and L-shaped domains and we will compare the same variables as in the previous Section, i.e. the number of the Ossen iterations, the number of matrix multiplications and the time requirement. A complete description of Ossen's iterations for this problem is in Section 4.4.

Let us choose the parameters $tol = 10^{-4}$, $ssntol = 10^{-5}$, $r_{tol} = 0.9$ and $c_{fact} = 0.9$. The maximum nmber of Ossen, Newton, BiCGstab iterations is 100,100, 1000, respectively. This settings will apply to all subsequent experiments.

Squared domain

Let's start with the same domain and the right-hand side $\boldsymbol{f}$ as in Section 5.1, but one part of the boundary γ_S will be with stick-slip boundary condition, witch is shown in Figure 15. The resulting velocity and pressure field is shown in Figure 16 and 17.

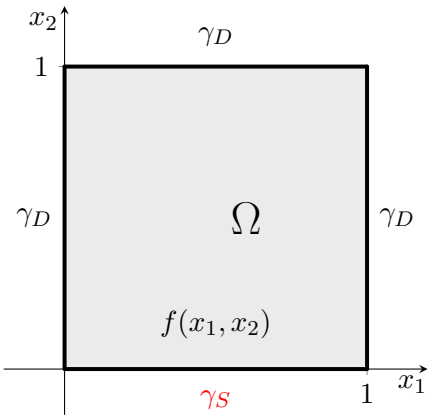


Figure 15: Squared domain with DNS conditions

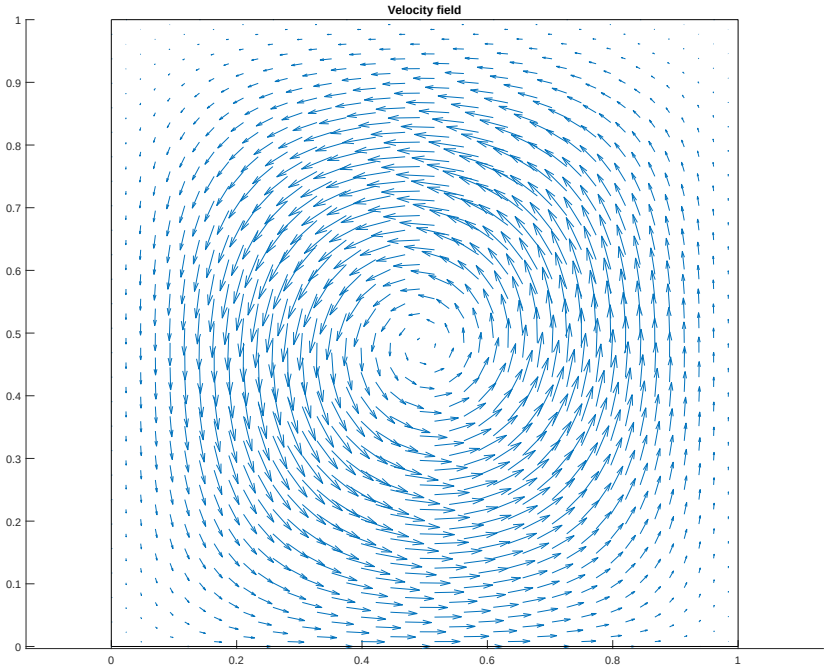


Figure 16: Velocity field

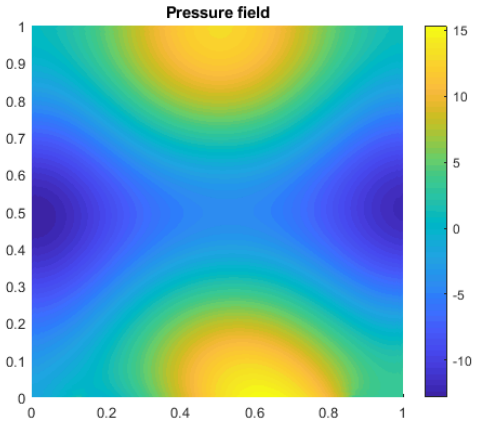


Figure 17: Velocity field

The effect of the stick-slip boundary condition for $\nu = 0.902$, $g = 0.1$, and $\kappa = 0.02$ is further shown in Figure 19, where $\mathbf{u} = \sqrt{\mathbf{u}_{x_1}^2 + \mathbf{u}_{x_2}^2}$ and the course of the stick-slip condition is shown in Figure 18, whrere the red curve is the stress and blue is the tangential speed.

The transition between the stick and slip condition is clear for $\nu = 0.902$, $g = 5$, and $\kappa = 0.2$ and 10 which is shown in Figures 20-23. The effect of κ on the stick-slip boundary condition is clear from Figures 20 and 22 for $\kappa = 0.02$ and 5.

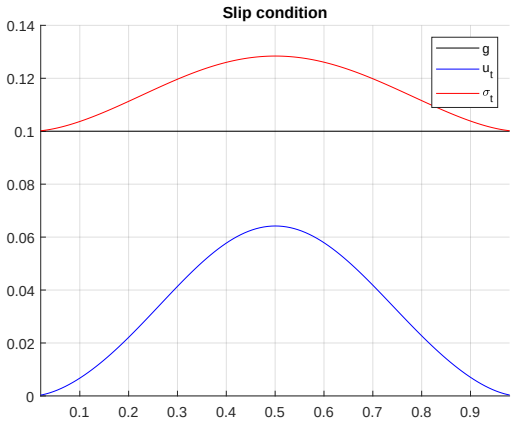


Figure 18: $g = 0.1$, $\kappa = 0.02$

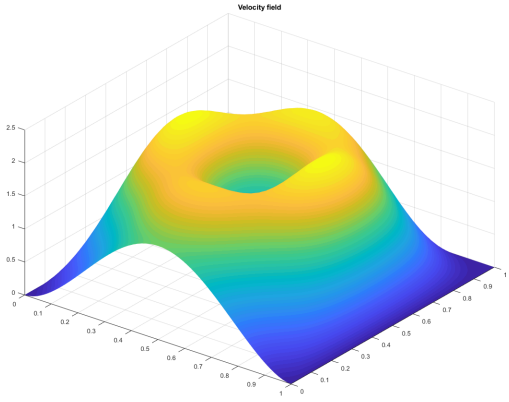


Figure 19: p_1 and p_2 for $\nu = 0.001$.

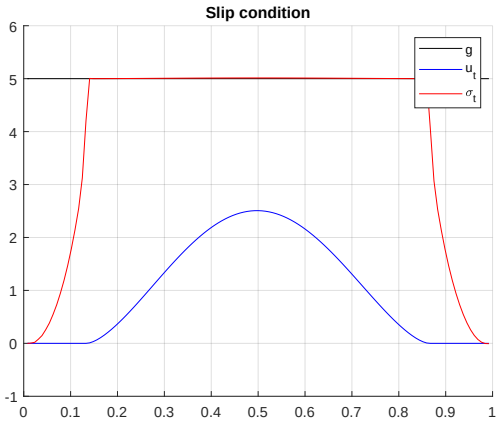


Figure 20: $g = 5$, $\kappa = 0.02$

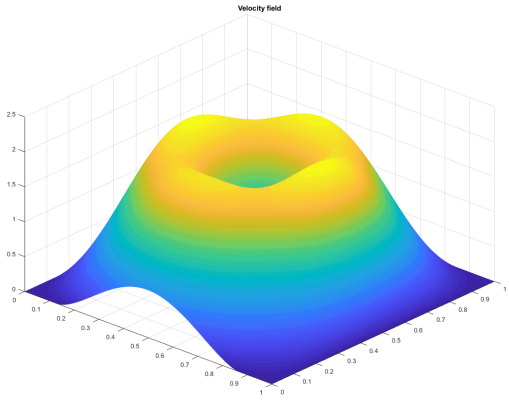


Figure 21: p_1 and p_2 for $\nu = 0.001$.

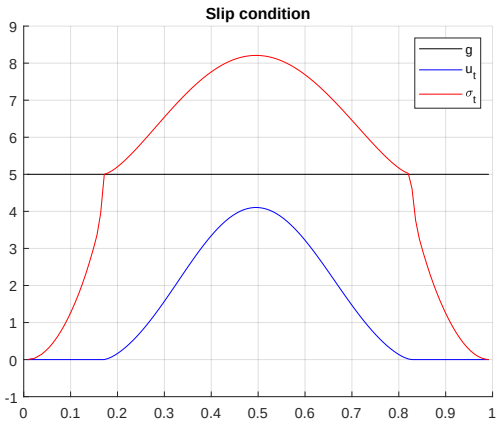


Figure 22: $g = 5$, $\kappa = 10$

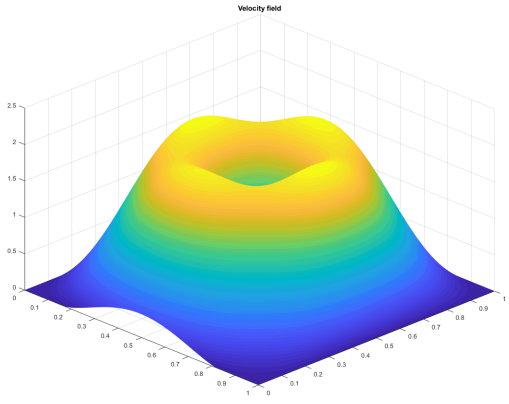


Figure 23: p_1 and p_2 for $\nu = 0.001$.

The test results of the various preconditioners are shown in Tables 11-12. It can be seen that the mass matrix preconditiner has worse results than non-preconditioned BiCGstab and for bigger Re numbers it does not even converge anymore or the solution time was very big. So for next experiments on the rectangle and L-shaped domain we will tests only clear BiCGstab and BiCGstab with diagonal preconditioners which has good performance results. It is obvious that even here the diagonal preconditioner has the best performance and converges even when clear BiCGstab no longer.

Table 11: DNS squared domain $\nu = 0.902$

Solver	BiCGstab		Mass matrix		Diagonal	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
9/8	1/5	0.01	1/7	0.00	1/7	0.00
25/32	3/205	0.02	3/173	0.02	3/107	0.01
81/128	3/294	0.04	3/377	0.05	3/123	0.02
289/512	3/457	0.11	3/590	0.15	4/144	0.07
1089/2048	3/373	0.44	3/4896	4.29	4/205	0.36
4225/8192	3/615	4.89	3/2149	15.59	4/209	2.56
16641/32768	3/549	30.12	3/11949	499.80	4/253	21.53

Table 12: DNS squared domain $\nu = 0.00902$

Solver	BiCGstab		Mass matrix		Diagonal	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)	it/n_{MV}	time(s)
9/8	1/5	0.00	-/-	-	1/5	0.00
25/32	7/388	0.04	-/-	-	7/448	0.05
81/128	4/556	0.06	-/-	-	6/651	0.08
289/512	4/1309	0.26	-/-	-	6/743	0.22
1089/2048	4/3395	2.96	-/-	-	7/1177	1.40
4225/8192	-/-	-	-/-	-	6/1493	12.10
16641/32768	-/-	-	-/-	-	7/1677	102.25

Furthermore, the dependence of the number of iterations it, number of matrix multiplication n_{MV} and time requirement on g and κ changes for the diagonal preconditioner was tested and it was observed that in all cases the dependence on g and κ is weak and relatively random. Only the time requirement grows regularly for grows g, but only in units of percent.

Rectangular domain

Let's have the rectangular domain $\Omega = (0, 5) \times (0, 1)$ which is in the Figure 24. We will perform the experiments with BiCGstab and BiCGstab with diagonal preconditioner and we will monitor the same value as in the squared domain. Let $\boldsymbol{f} = \mathbf{0}$, $\nu = 0.902$ and 0.00902, $g = 0.5$ and 0.005, $\kappa = 10$, and $u_{in} = 1$.

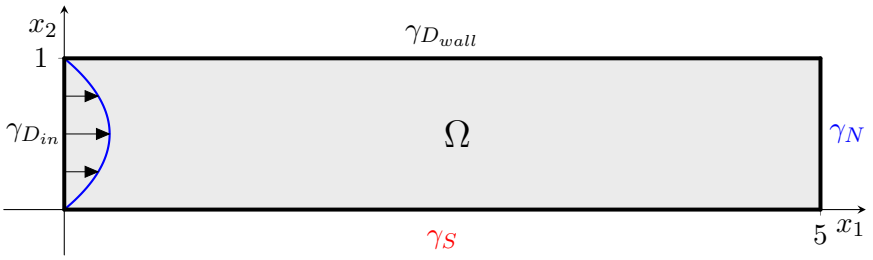


Figure 24: Rectangular domain with DNS conditions

Graphical results are in Figure 25 and 26. The results are shown in Tables 13-14. It is obvious that even here the diagonal preconditioner has the best performance.

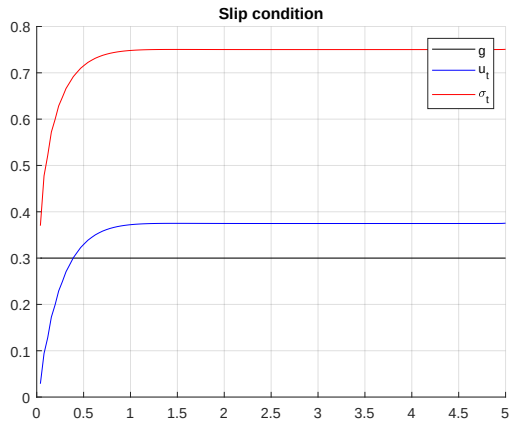


Figure 25: $g = 0.3$, $\kappa = 10$, $vel = 1$, $\nu = 0.902$

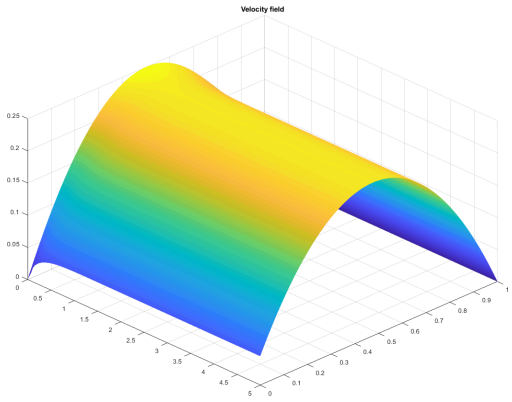


Figure 26: Velocity field

Table 13: DNS rectangular domain $\nu = 0.902$

Solver	BiCGstab		Diagonal	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)
9/8	2/88	0.01	3/100	0.02
25/32	3/184	0.02	3/217	0.02
81/128	2/362	0.04	2/200	0.04
289/512	2/529	0.12	2/282	0.08
1089/2048	2/1491	1.37	2/352	0.43
4225/8192	2/1298	9.86	2/437	3.73
16641/32768	2/4927	206.32	2/535	27.38

Table 14: DNS rectangular domain $\nu = 0.00902$

Solver	BiCGstab		Diagonal	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)
9/8	3/83	0.01	3/165	0.02
25/32	3/370	0.03	4/208	0.03
81/128	3/604	0.06	3/271	0.04
289/512	2/688	0.14	3/305	0.10
1089/2048	2/1494	1.34	3/311	0.46
4225/8192	2/1891	14.02	3/349	3.40
16641/32768	2/1823	80.47	2/435	23.16

L-shaped domain

Next, we will have L-shaped domain $\Omega = (0,5) \times (0,2) \backslash (0,1) \times (0,1)$ which is in Figure 27.

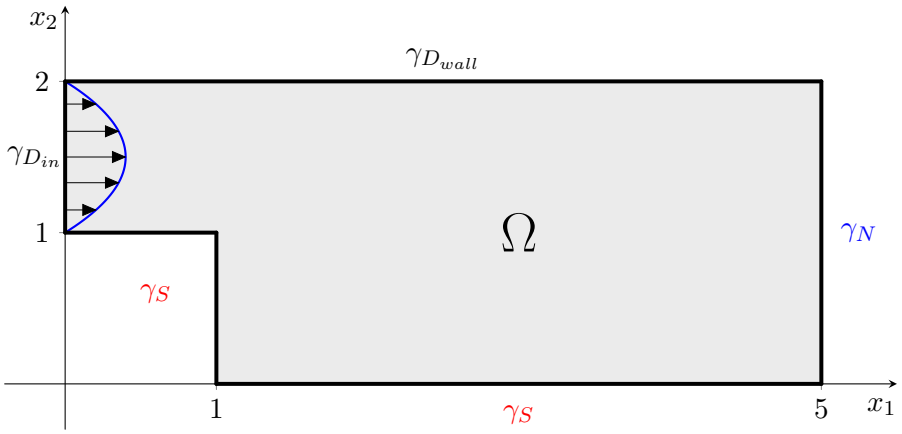


Figure 27: L-shaped domain with DNS conditions

We will perform the same experiments as on the rectangular domain. Let $\boldsymbol{f} = \mathbf{0}$, $\nu = 0.902$ and 0.00902, $g = 0.1$ and 0.001, $\kappa = 10$ and $u_{in} = 1$. The results are shown in Tables 15-16. It is obvious that even here the diagonal preconditioner has the best performance. It can be see, that for bigger Re number has BiCGstab without preconditioner worst stability.

Table 15: DNS L-shaped $\nu = 0.902$

Solver	BiCGstab		Diagonal	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)
174/278	3/975	0.15	3/295	0.07
206/333	3/1434	0.27	3/236	0.07
744/1332	3/2943	1.38	3/350	0.27
2819/5328	3/6673	21.61	3/394	1.82
10965/21312	2/4325	89.39	3/695	17.29

Table 16: DNS L-shaped $\nu = 0.00902$

Solver	BiCGstab		Diagonal	
n_p/n_t	it/n_{MV}	time(s)	it/n_{MV}	time(s)
174/278	6/3862	0.52	6/789	0.17
206/333	6/3741	0.62	6/856	0.19
744/1332	5/8458	3.82	7/1140	0.80
2819/5328	5/23668	76.44	8/1372	5.95
10965/21312	5/284109	5712.67	8/1794	45.21

The graphical results are in Figure 28 and 29.

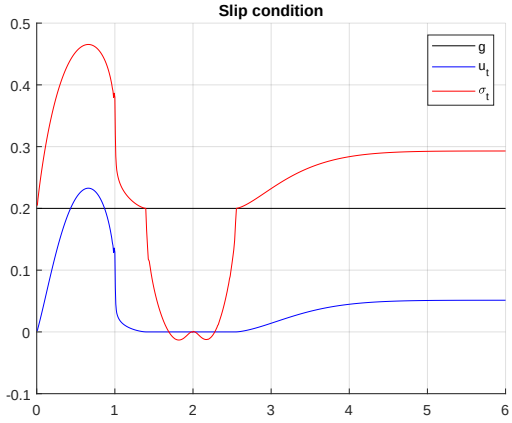


Figure 28: $g = 0.2$, $\kappa = 1$, $vel = 2$, $\nu = 0.902$

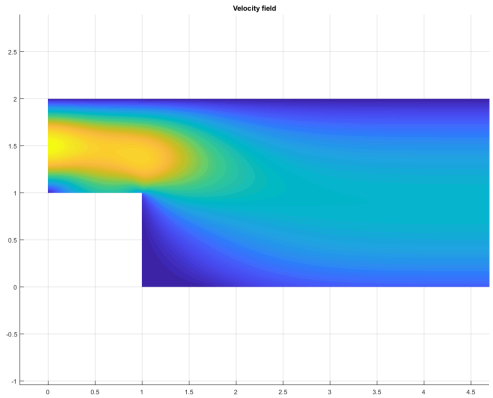


Figure 29: Velocity field

6 Numerical experiments in three dimensions

In this section we will perform numerical experiment in three dimensions. We will test the convergence of the Ossen iterations for the Navier-Stokes problem with the Dirichlet boundary condition in three space dimensions for which we know the solution. This will verify the correctness of the function for assembling matrices described in Code 4 which is derived in Sections A and C.

6.1 Convergence of the Ossen iterations

Let us have the similar problem as in two space dimensions. We will solve the Navier-Stokes equation as follows:

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p &= \boldsymbol{f} && \text{in } \Omega, \\ \nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{u}_D && \text{on } \partial\Omega, \end{aligned} \tag{6.1}$$

where we consider the domain $\Omega = (0,1)^3$ with the Dirichlet boundary condition $\mathbf{u}_D = 0$ on $\partial\Omega$, $\boldsymbol{u} : \overline{\Omega} \to \mathbb{R}^3$, $p : \overline{\Omega} \to \mathbb{R}$, and viscosity $\nu = 1$. Let $\boldsymbol{x} = [x, y, z]^\mathsf{T}$. The right-hand side $\boldsymbol{f}$ is adjusted such that analytical solution is as follow:

$$\begin{aligned} u_{ex}(x,y,z) &= 4z(1-z)\sin(2\pi y)\left(1-\cos(2\pi x)\right), \\ u_{ey}(x,y,z) &= 4z(1-z)\sin(2\pi x)(\cos(2\pi y)-1), \\ u_{ez}(x,y,z) &= 0, \\ p_e(x,y,z) &= 2\pi\left(\cos(2\pi y)-\cos(2\pi x)-\cos(2\pi z)\right), \end{aligned}$$

where u_{ex}, u_{ey} and u_{ez} represent the velocity in x, y, and z direction, respectively. We will linearize the convective term $\boldsymbol{u}\cdot\nabla\boldsymbol{u}$ in (6.1) as in the 2D case, thus obtaining formally the same Ossen problem 5.2 and the same linear system 5.3 as in Section 5.1, which we will solve by the simpler version of the Ossen iterations algorithm, which are also desribed in Section 5.1.

We will test the order of the convergence for the similar norms as in two space dimensions, i.e.

$$\begin{aligned} e_1(h) &= ||\boldsymbol{u}_h - \boldsymbol{u}_{exp}||_{(L^2(\Omega))^3}, \\ e_2(h) &= ||p_h - p_{exp}||_{L^2(\Omega)} + ||\boldsymbol{u}_h - \boldsymbol{u}_{exp}||_{(H^1(\Omega))^3}. \end{aligned}$$

The order of the convergence $p \in \mathbb{R}$ is also defined as follows:

$$p_{i,j} := \log_2\left(\frac{e_i(h_{j-1})}{e_i(h_j)}\right), \qquad i = 1, 2. \tag{6.2}$$

The results are in Table 17 and graphical comparsion is in Figure 30.

Table 17: p_1 and p_2 for $\nu = 1$.

h_j	e_1	p_1	e_2	p_2
1/4	0.25493	-	6.6601	-
1/8	0.08602	1.5673	3.1494	1.0805
1/16	0.02211	1.9598	1.2586	1.3232
1/32	0.00561	1.9781	0.5322	1.2418

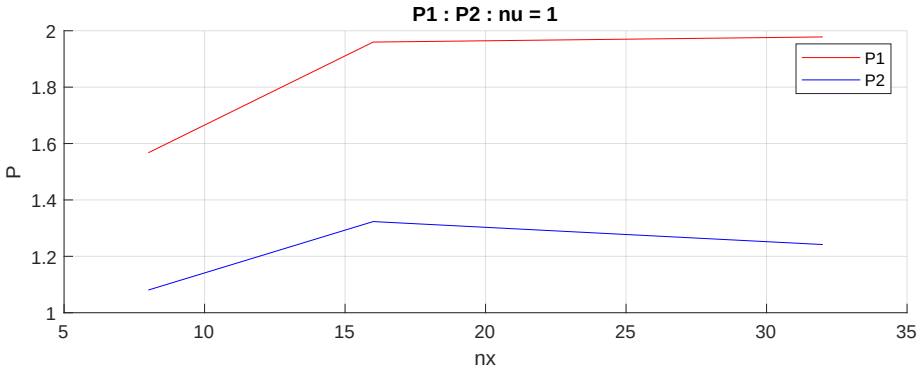


Figure 30: p_1 and p_2 for $\nu = 1$.

The results indicate that for corresponding approximation the following relations apply:

$$\lim_{h_j \to 0} p_1 = 2,$$

$$\lim_{h_j \to 0} p_2 = 1,$$

which is as in the 2D case consistent with the experimental observations on superconvergence of FEM for MINI elements (P1b/P1) for the Stokes problem [9]. In our case we observe superconvergence result for the Navier-Stokes problem in three space dimensions.

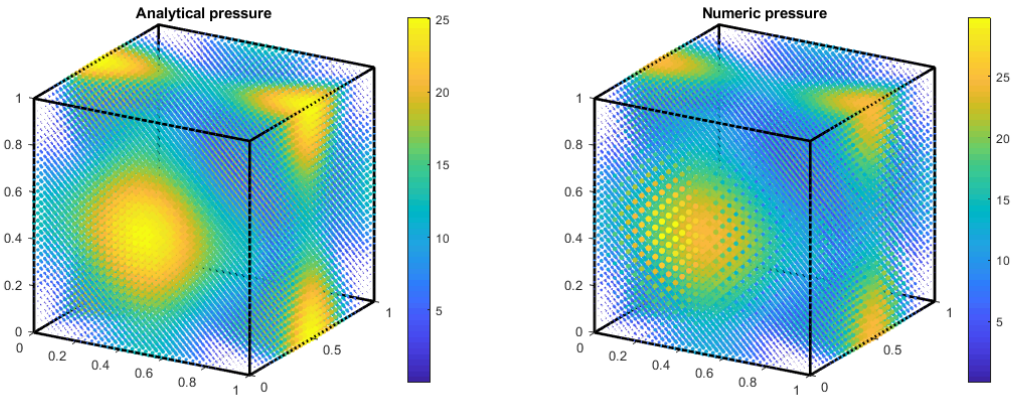


Figure 31: Pressure field for $\nu = 1$

Figure 31 show analytical and numerical pressure field. Figure 32 shows that the error between them is relatively small except for the nodes at the boundary $\partial\Omega$, and the graphical representation of the velocity field is shown in Figure 33.

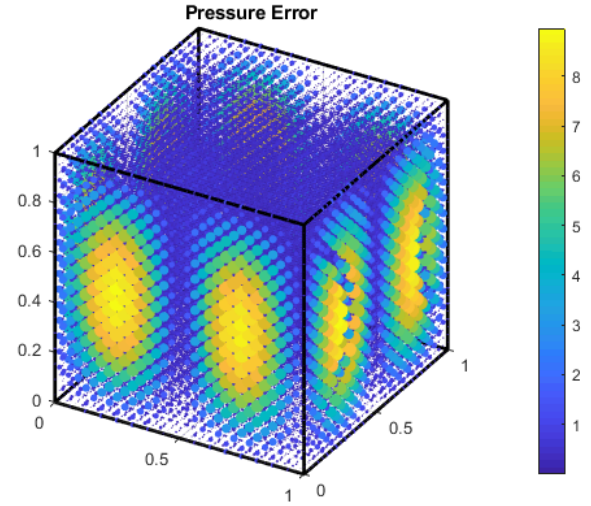


Figure 32: Pressure field error for $\nu = 1$

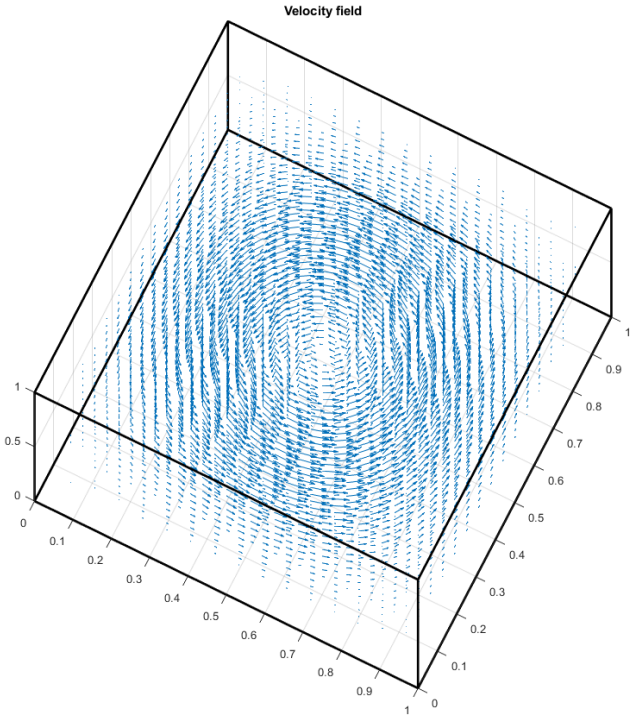


Figure 33: Velocity field for $\nu = 1$

7 Conclusion

The diploma thesis deals with the solution of the Navier-Stokes problem with a stick-slip boundary condition. First, the problem was described, its weak formulation was derived and its linearization by the Ossen iterations was proposed. Furthermore, the problem was discretizet by the mixed finite element method with the P1-bubble/P1 elements. The resulting problem was solved by the Ossen iterations, which linearized the convective term. The second nonlinearity caused by the stick-slip boundary condition was solved by the semi-smooth Newton method. Thanks to the convective term, the linear system is nonsymmetric, so the BiCGstab algorithm was chosen as the internal solver of the semi-smooth Newton method.

Experiments in two space dimensions were then described. First, the convergence of the Ossen iterations for the Navier-Stokes problem was verified on the example with zero Dirichlet boundary conditions with a known solution. Superlinearity was found, which has already been described for the Stokes problem in the paper [9] and was also verified here for the Navier-Stokes problem. After that, we went to test varrious preconditioners for the problem with the Dirichlet and the Neumann boundary conditions and finally to the problem with the stick-slip boundary conditions. For the problem with stick-slip boundary condition, its behavior for various settings was investigated. Finally, the convergence was verified for the problem in three space dimensions. Superlinearity was also verified here.

Within the work, non-vectorized and vectorized codes were derived for the assembly of the stiffness matrices and the right-side vector for the Navier-Stokes problem with the P1-bubble/P1 elements for two and three space dimensions. Detailed derivations of the algorithms inspired by the procedures described in article [7] for the Stokes system and the implemented codes can be found in Appendix A-C.

It should also be noted that in the future, preconditioners for the Shur complement arising from the Stokes and the Ossen linear systems in article [10] will also be implemented.

References

[1] Djoko, J.K. & Mbehou, M.. (2013). Finite element analysis for Stokes and Navier-Stokes equations driven by threshold slip boundary conditions. international journal of numerical analysis and modeling, series B. 4. 235-255.

[2] FUJITA, Hiroshi. (1994). A mathematical analysis of motions of viscous incompressible fluid under leak or slip boundary conditions(Mathematical Fluid Mechanics and Modeling). RIMS Kōkyūroku. 888. 199-216.

[3] Navier, C.L.M.H.. (1823). Sur les lois du mouvement des fluides, Mem. Acad. R. Sci. Inst. France, 389-440.

[4] M.K. Gdoura. (2011). *Problème de Stokes avec des conditions aux limites non-linéaires: analyse numérique et algorithmes de résolution*, Thèse en co-tutelle, Université Tunis El Manar et Université de Caen Basse Normandie

[5] J. Pacholek. (2017). Semi-smooth Newton method for solving the Stokes equations with monotonously increasing slip condition, Master's thesis, VŠB-TU Ostrava, Czech Republic.

[6] Arnold, Douglas & Brezzi, Franco & Fortin, Michel. (1984). A stable finite element for the Stokes equations. Calcolo. 21. 337-344. 10.1007/BF02576171.

[7] Koko, J. (2019). Efficient MATLAB Codes for the 2D/3D Stokes Equation with the Mini-Element. Informatica, 30(2), 243-268. doi:10.15388/Informatica.2019.205

[8] Arzt, V. (2019). Finite element meshes and assembling of stiffness matrices, Bachelor's thesis, VŠB-TU Ostrava, Czech Republic.

[9] Cioncolini, Andrea & Boffi, Daniele. (2019). The MINI mixed finite element for the Stokes problem: An experimental investigation. Computers & Mathematics with Applications, Volume 77, Issue 9, 2019, Pages 2432-2446,

[10] Olshanskii, Maxim & Vassilevski, Yuri. (2007). Pressure Schur Complement Preconditioners for the Discrete Oseen Problem. SIAM Journal on Scientific Computing. 29. 2686-2704. 10.1137/070679776.

A Weak formulation of the problem with the convective term

Let us derive vectorized codes for assembling stiffness matrices for the Ossen problem without boudnary conditions. We will formulate the problem for two and three spatial dimensions.

Let $\Omega \subset \mathbb{R}^d,\ d = 2, 3$, is the bounded domain with a sufficiently smooth boundary $\partial\Omega$. We will find a vector function $\boldsymbol{u} : \overline{\Omega} \to \mathbb{R}^d$ and a scalar function $p : \overline{\Omega} \to \mathbb{R}$ satisfying the following system of partial differential equations:

$$-\nu\Delta\boldsymbol{u} + \boldsymbol{w}\cdot\nabla\boldsymbol{u} + \nabla p + \alpha\boldsymbol{u} = \boldsymbol{f} \qquad \text{in } \Omega, \tag{A.1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{A.2}$$

where $\boldsymbol{w} : \overline{\Omega} \to \mathbb{R}^d$ is given a sufficiently smooth function, $\nu > 0$ is the dynamic viscosity, $\alpha \geq 0$ is a parameter related to the time problem and $\boldsymbol{f} : \overline{\Omega} \to \mathbb{R}^d$ are forces acting on the fluid. Notice that the convective term looks like for $d = 2$, when $\boldsymbol{u} = (u_1, u_2)^\mathsf{T}$ and $\boldsymbol{w} = (w_1, w_2)^\mathsf{T}$:

$$\boldsymbol{w}\cdot\nabla\boldsymbol{u} = \begin{bmatrix} w_1u_{1x} + w_2u_{1y} \\ w_1u_{2x} + w_2u_{2y} \end{bmatrix}, \tag{A.3}$$

and for $d = 3$, when $\boldsymbol{u} = (u_1, u_2, u_3)^\mathsf{T}$ and $\boldsymbol{w} = (w_1, w_2, w_3)^\mathsf{T}$ as follows:

$$\boldsymbol{w}\cdot\nabla\boldsymbol{u} = \begin{bmatrix} w_1u_{1x} + w_2u_{1y} + w_3u_{1z} \\ w_1u_{2x} + w_2u_{2y} + w_3u_{2z} \\ w_1u_{3x} + w_2u_{3y} + w_3u_{3z} \end{bmatrix}. \tag{A.4}$$

The weak formulation of the homogeneous Dirichlet problem of (A.1)-(A.2) reads as follow:

$$\left.\begin{aligned} &\text{Find } \boldsymbol{u} \in (H_0^1(\Omega))^d,\ p \in L^2(\Omega) \text{ such that} \\ &\nu\int_\Omega \nabla\boldsymbol{u} : \nabla\boldsymbol{v} + \int_\Omega (\boldsymbol{w}\cdot\nabla\boldsymbol{u})\cdot\boldsymbol{v} + \alpha\int_\Omega \boldsymbol{u}\cdot\boldsymbol{v} - \int_\Omega p(\nabla\cdot\boldsymbol{v}) = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v} \quad \forall\boldsymbol{v} \in (H_0^1(\Omega))^d \\ &\int_\Omega q(\nabla\cdot\boldsymbol{u}) = 0 \quad \forall q \in L^2(\Omega), \end{aligned}\right\} \ (P)$$

where the first integral

$$\nu\int_T \nabla\boldsymbol{u} : \nabla\boldsymbol{v} = \nu\int_T \nabla\boldsymbol{u}_1\cdot\nabla\boldsymbol{v}_1 + \ldots + \nu\int_T \nabla\boldsymbol{u}_d\cdot\nabla\boldsymbol{v}_d$$

will be represented by a stiffness matrix $\mathbb{R}$, the second integral by a convective matrix $\mathbb{C}$ and the third by a mass matrix $\mathbb{M}$. The last integral in the first equation and the integral in the second equation will by represented by a divergence matrix $\mathbb{B}$ and the integral on the right side in first equation by a vector of the right side $\mathbf{b}$. We will describe all these members in the following chapters for two and three dimensions and introduce non-vectorized and vectorized codes to assembly them.

B Assembly function in two dimensions

Let us have the triangulation $\mathcal{T}_h$ of the domain $\Omega \subset \mathbb{R}^2$. On the triangle $T \in \mathcal{T}_h$ with vertex $\mathbf{p}_i = [x_i, y_i]^\mathsf{T},\ i = 1, 2, 3$ we will have three nonzero linear basis function $\phi_1(\boldsymbol{x})$, $\phi_2(\boldsymbol{x})$, $\phi_3(\boldsymbol{x}) \in P_1(T)$, $\boldsymbol{x} = [x, y]^\mathsf{T} \in T$ defined by conditions: $\phi_i(\mathbf{p}_j) = \delta_{ij},\ i = 1, 2, 3$ and the bubble basis function, which is defined on the triangle T as follows:

$$\phi_b(\boldsymbol{x}) = 3^3 \cdot \phi_1(\boldsymbol{x}) \cdot \phi_2(\boldsymbol{x}) \cdot \phi_3(\boldsymbol{x}), \quad \boldsymbol{x} \in T.$$

For simplicity, we will denote $\phi_4 = \phi_b$. We will aproximate the components of the velocity vector $\boldsymbol{u}^h = (u_1^h, u_2^h)^\mathsf{T}$ and the pressure p^h on T as follows:

$$u_k^h = \sum_{j=1}^{4} u_{kj}\phi_j, \quad k = 1, 2,$$
$$p^h = \sum_{j=1}^{3} p_j\phi_j.$$

Mass matrix

First, we derive the mass matrix from the integral:

$$\alpha \int_T \boldsymbol{u}^h \cdot \boldsymbol{v}^h = \alpha \int_T u_1^h v_1^h + \alpha \int_T u_2^h v_2^h,$$

where $\boldsymbol{v}^h = (v_1^h, v_2^h)^\mathsf{T}$. For $\boldsymbol{v}^h = (\phi_i, 0)^\mathsf{T}$ and $\boldsymbol{v}^h = (0, \phi_i)^\mathsf{T}$ we get:

$$\sum_{j=1}^{4} u_{1j}\alpha \int_T \phi_j\phi_i, \quad i = 1, \ldots, 4,$$

$$\sum_{j=1}^{4} u_{2j}\alpha \int_T \phi_j\phi_i, \quad i = 1, \ldots, 4,$$

respectively. The local mass matrix $\mathbb{M}_T \in \mathbb{R}^{8\times 8}$ therefore form as follow:

$$\mathbb{M}_T = \left[\begin{array}{cc|cc} \mathbf{M}_T & \mathbf{m}_T & 0 & 0 \\ \mathbf{m}_T^\mathsf{T} & \omega_M & 0 & 0 \\ \hline 0 & 0 & \mathbf{M}_T & \mathbf{m}_T \\ 0 & 0 & \mathbf{m}_T^\mathsf{T} & \omega_M \end{array}\right],$$

where $\mathbf{M}_T,\ \mathbf{m}_T,\ \omega_M$ are derived in [8] and read as follows:

$$\mathbf{M}_T = \frac{\alpha|T|}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \quad \mathbf{m}_T = \frac{3\alpha|T|}{20}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \omega_M = \frac{81\alpha|T|}{280}.$$

Stiffness matrix

Next, let us derive the local stiffness matrix $\mathbb{R}_T \in \mathbb{R}^{8\times 8}$. The respective integral has the following form:

$$\begin{aligned}\nu\int_T \nabla \boldsymbol{u}^h : \nabla \boldsymbol{v}^h &= \nu\int_T \nabla \boldsymbol{u}_1^h \cdot \nabla \boldsymbol{v}_1^h + \nu\int_T \nabla \boldsymbol{u}_2^h \cdot \nabla \boldsymbol{v}_2^h = \\ &= \nu\int_T u_{1x}^h v_{1x}^h + u_{1y}^h v_{1y}^h + u_{2x}^h v_{2x}^h + u_{2y}^h v_{2y}^h.\end{aligned}$$

For $\boldsymbol{v}^h = (\phi_i, 0)^\mathsf{T}$ we get:

$$\sum_{j=1}^{4} u_{1j}\nu\int_T (\phi_{jx}\phi_{ix} + \phi_{jy}\phi_{iy}), \quad \text{for } i = 1,\dots,4,$$

and for $\boldsymbol{v}^h = (0, \phi_i)^\mathsf{T}$ we get:

$$\sum_{j=1}^{4} u_{2j}\nu\int_T (\phi_{jx}\phi_{ix} + \phi_{jy}\phi_{iy}), \quad \text{for } i = 1,\dots,4.$$

The local stiffness matrix $\mathbb{R}_T \in \mathbb{R}^{8\times 8}$ have the form as follows:

$$\mathbb{R}_T = \left[\begin{array}{cc|cc} \mathbf{R}_T & \mathbf{r}_T & 0 & 0 \\ \mathbf{r}_T^\mathsf{T} & \omega_R & 0 & 0 \\ \hline 0 & 0 & \mathbf{R}_T & \mathbf{r}_T \\ 0 & 0 & \mathbf{r}_T^\mathsf{T} & \omega_R \end{array}\right],$$

where

$$\begin{aligned}(\mathbf{R}_T)_{ij} &= \nu\int_T (\phi_{jx}\phi_{ix} + \phi_{jy}\phi_{iy}), \quad i,j = 1,2,3, \\ \omega_R &= \nu\int_T (\phi_{bx}\phi_{bx} + \phi_{by}\phi_{by}), \\ (\mathbf{r}_T)_j &= \nu\int_T (\phi_{jx}\phi_{bx} + \phi_{jy}\phi_{by}) = 0, \quad j = 1,2,3.\end{aligned}$$

Let $\mathbf{p}_1 = [x_1, y_1]^\mathsf{T}$, $\mathbf{p}_2 = [x_2, y_2]^\mathsf{T}$ and $\mathbf{p}_3 = [x_3, y_3]^\mathsf{T}$ are the vertices of the triangle T. Let us denote:

$$\boldsymbol{x}_T = \begin{bmatrix} x_3 - x_2 \\ x_1 - x_3 \\ x_2 - x_1 \end{bmatrix} = \begin{bmatrix} x_{32} \\ x_{13} \\ x_{21} \end{bmatrix}, \quad \boldsymbol{y}_T = \begin{bmatrix} y_2 - y_3 \\ y_3 - y_1 \\ y_1 - y_2 \end{bmatrix} = \begin{bmatrix} y_{23} \\ y_{31} \\ y_{12} \end{bmatrix}.$$

In [8] it is proved that the following holds:

$$\boldsymbol{y}_T = 2|T|[\phi_{1x}, \phi_{2x}, \phi_{3x}]^\mathsf{T}, \quad \boldsymbol{x}_T = 2|T|[\phi_{1y}, \phi_{2y}, \phi_{3y}]^\mathsf{T},$$

and we get:

$$\mathbf{R}_T = \frac{\nu}{4|T|}(\boldsymbol{y}_T\boldsymbol{y}_T^\mathsf{T} + \boldsymbol{x}_T\boldsymbol{x}_T^\mathsf{T}),$$
$$\omega_R = \frac{81\nu}{40|T|}(y_{T1}^2 + x_{T1}^2 - y_{T2}y_{T3} - x_{T2}x_{T3}).$$

Convective matrix

The respective integrals for the convective term (A.3) for $d = 2$ has form as follow:

$$\int_T (\boldsymbol{w}\cdot\nabla\boldsymbol{u}^h)\cdot\boldsymbol{v}^h = \int_T \left(\begin{bmatrix} w_1u_{1x}^h + w_2u_{1y}^h \\ w_1u_{2x}^h + w_2u_{2y}^h \end{bmatrix} \cdot [v_1^h, v_2^h] \right) =$$
$$= \int_T (w_1u_{1x}^hv_1^h + w_2u_{1y}^hv_1^h) + \int_T (w_1u_{2x}^hv_2^h + w_2u_{2y}^hv_2^h).$$

For $\boldsymbol{v}^h = (\phi_i, 0)^\mathsf{T}$ we get:

$$\sum_{j=1}^4 u_{1j}\int_T (w_1\phi_{jx}\phi_i + w_2\phi_{jy}\phi_i), \quad \text{for } i = 1,\dots,4,$$

and for $\boldsymbol{v}^h = (0, \phi_i)^\mathsf{T}$ we get:

$$\sum_{j=1}^4 u_{2j}\int_T (w_1\phi_{jx}\phi_i + w_2\phi_{jy}\phi_i), \quad \text{for } i = 1,\dots,4.$$

The local convective matrix $\mathbb{C}_T \in \mathbb{R}^{8\times 8}$ have the form as follows:

$$\mathbb{C}_T = \left[\begin{array}{cc|cc} \mathbf{C}_T & \mathbf{c}_{uT} & 0 & 0 \\ \mathbf{c}_{lT}^\mathsf{T} & \omega_C & 0 & 0 \\ \hline 0 & 0 & \mathbf{C}_T & \mathbf{c}_{uT} \\ 0 & 0 & \mathbf{c}_{lT}^\mathsf{T} & \omega_C \end{array}\right].$$

In [8] is proved that

$$\int_T \phi_i = \frac{2|T|}{6}, \quad i = 1,2,3.$$

Let us denote the approximation of $\boldsymbol{w}$ as follows:

$$w_k^{(T)} = \frac{1}{3}(w_{k,1} + w_{k,2} + w_{k,3}), \quad k = 1,2,$$

where $w_{k,i}$ represents value at the *i-th* vertex of the triangle T, then for $i, j = 1, 2, 3$,

$$(\mathbf{C}_T)_{ij} := w_1^{(T)} \phi_{jx} \int_T \phi_i + w_2^{(T)} \phi_{jy} \int_T \phi_i = w_1^{(T)} y_{Tj} \frac{1}{2|T|} \frac{2|T|}{6} + w_2^{(T)} x_{Tj} \frac{1}{2|T|} \frac{2|T|}{6}.$$

We get:

$$\mathbf{C}_T = w_1^{(T)} \frac{1}{6} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{y}_T^\mathsf{T} + w_2^{(T)} \frac{1}{6} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{x}_T^\mathsf{T}.$$

The bubble component w_C has form as follows:

$$\begin{aligned} \omega_C :=& \int_T w_1^{(T)} \phi_{bx} \phi_b + w_2^{(T)} \phi_{by} \phi_b = \\ =& 27^2 w_1^{(T)} \left[\phi_{1x} \int_T \phi_1 \phi_2^2 \phi_3^2 + \phi_{2x} \int_T \phi_1^2 \phi_2 \phi_3^2 + \phi_{3x} \int_T \phi_1^2 \phi_2^2 \phi_3 \right] + \\ &+ 27^2 w_2^{(T)} \left[\phi_{1y} \int_T \phi_1 \phi_2^2 \phi_3^2 + \phi_{2y} \int_T \phi_1^2 \phi_2 \phi_3^2 + \phi_{3y} \int_T \phi_1^2 \phi_2^2 \phi_3 \right], \end{aligned}$$

where all subintegral are equal:

$$\int_T \phi_1 \phi_2^2 \phi_3^2 = \int_T \phi_1^2 \phi_2 \phi_3^2 = \int_T \phi_1^2 \phi_2^2 \phi_3 = \frac{2|T|}{1260}.$$

Then we can write that

$$\omega_C = \frac{81}{140} w_1^{(T)} [1, 1, 1] \cdot \boldsymbol{y}_T + \frac{81}{140} w_2^{(T)} [1, 1, 1] \cdot \boldsymbol{x}_T = 0,$$

where the sum of the components $x_{T1} + x_{T2} + x_{T3} = 0$ as well as sums of $\mathbf{y}_T$.

Next we describe $\mathbf{c}_{uT}$:

$$(\mathbf{c}_{uT})_i = \int_T w_1 \phi_{bx} \phi_i + w_2 \phi_{by} \phi_i, \quad i = 1, 2, 3.$$

First we calculate the integrals:

$$\int_T \phi_1 \phi_2 \phi_3 = \frac{2|T|}{120}, \qquad \int_T \phi_1^2 \phi_3 = \int_T \phi_1^2 \phi_2 = \frac{2|T|}{60},$$

then for $i = 1$:

$$\begin{aligned} (\mathbf{c}_{uT})_1 = \int_T w_1 \phi_{bx} \phi_1 + w_2 \phi_{by} \phi_1 &= 27 w_1^{(T)} \left[\phi_{1x} \int_T \phi_1 \phi_2 \phi_3 + \phi_{2x} \int_T \phi_1^2 \phi_3 + \phi_{3x} \int_T \phi_1^2 \phi_2 \right] + \\ &+ 27 w_2^{(T)} \left[\phi_{1y} \int_T \phi_1 \phi_2 \phi_3 + \phi_{2y} \int_T \phi_1^2 \phi_3 + \phi_{3y} \int_T \phi_1^2 \phi_2 \right] = \\ = \frac{27}{120} w_1^{(T)} \left[y_{T1} + 2 y_{T2} + 2 y_{T3} \right] + \frac{27}{120} w_2^{(T)} \left[x_{T1} + 2 x_{T2} + 2 x_{T3} \right] &= \\ = \frac{9}{40} w_1 [1, 2, 2] \cdot \mathbf{y}_T + \frac{9}{40} w_2 [1, 2, 2] \cdot \mathbf{x}_T &= \frac{9}{40} [1, 2, 2] \cdot (w_1 \mathbf{y}_T + w_2 \mathbf{x}_T). \end{aligned}$$

Let us denote $\mathbf{c}_1 = [1,2,2]$, $\mathbf{c}_2 = [2,1,2]$ and $\mathbf{c}_3 = [2,2,1]$. It can by proved that

$$(\mathbf{c}_{uT})_i = \frac{9}{40}\mathbf{c}_i \cdot (w_1\mathbf{y}_T + w_2\mathbf{x}_T) \quad i = 1,2,3,$$

and finally

$$\mathbf{c}_{uT} = \frac{9}{40}\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}(w_1\boldsymbol{y}_T + w_2\boldsymbol{x}_T).$$

As the last member we describre $\mathbf{c}_{lT}$. For $j = 1,2,3$:

$$\begin{aligned}(\mathbf{c}_{lT})_j &= \int_T w_1\phi_{jb}\phi_b + w_2\phi_{jy}\phi_b = \\ &= 27w_1^{(T)}\phi_{jx}\int_T \phi_1\phi_2\phi_3 + 27w_2^{(T)}\phi_{jy}\int_T \phi_1\phi_2\phi_3 = \\ &= 27w_1^{(T)}\frac{1}{2|T|}y_{Tj}\frac{2|T|}{120} + 27w_2^{(T)}\frac{1}{2|T|}x_{Tj}\frac{2|T|}{120},\end{aligned}$$

where

$$\int_T \phi_1\phi_2\phi_3 = \frac{2|T|}{120}.$$

We get:

$$\mathbf{c}_{lT} = \frac{9}{40}\left(w_1^{(T)}\mathbf{y}_T^{\mathsf{T}} + w_2^{(T)}\mathbf{x}_T^{\mathsf{T}}\right).$$

Divergence and gradient matrix

Now, we explain the approximation of the integral with the pressure component in the first equation and the approximation of the second equation. First, we derive local divergence matrices for the second equation. We come out of the expression:

$$-\int_T q^h(u_{1x}^h + u_{2y}^h),$$

where we substitute the approximations u_1^h, u_2^h, and for q^h we gradually choose all the basis functions used in the pressure approximation. We will get:

$$\sum_{j=1}^{4} u_{1j}\left(-\int_T \phi_i\phi_{jx}\right) + \sum_{j=1}^{4} u_{2j}\left(-\int_T \phi_i\phi_{jy}\right), \quad i = 1,2,3.$$

The local divergence matrix $\mathbb{B}_T \in \mathbb{R}^{3\times 8}$ should be divided into four parts:

$$\mathbb{B}_T = [\mathbf{B}_{1T}, \mathbf{B}_{1bT}, \mathbf{B}_{2T}, \mathbf{B}_{2bT}],$$

where

$$\mathbf{B}_{1T} = -\frac{1}{6}[\boldsymbol{y}_T, \boldsymbol{y}_T, \boldsymbol{y}_T]^\mathsf{T}, \qquad \mathbf{B}_{1bT} = \frac{9}{40}\boldsymbol{y}_T,$$
$$\mathbf{B}_{2T} = -\frac{1}{6}[\boldsymbol{x}_T, \boldsymbol{x}_T, \boldsymbol{x}_T]^\mathsf{T}, \qquad \mathbf{B}_{2bT} = \frac{9}{40}\boldsymbol{x}_T,$$

which is proved in [8].

Finally, note the integral with the pressure component:

$$-\int_T p(\nabla \cdot \boldsymbol{v}) = -\int_T p(v_{1x} + v_{2x}),$$

where $\boldsymbol{v} = (v_1, v_2)^\mathsf{T}$. For $\boldsymbol{v} = (\phi_i, 0)$ and $\boldsymbol{v} = (0, \phi_i)$ we get:

$$\sum_{j=1}^{3} p_j(-\int_T \phi_j \phi_{ix}), \quad \sum_{j=1}^{3} p_j(-\int_T \phi_j \phi_{iy}), \qquad i = 1, \ldots, 4.$$

It can be seen that the local gradient matrix is a transposition of the local divergence matrix.

Vector of the right side

In the problem (P) it remains the local vector of the right side which arises from the approximation of the integral:

$$\int_T \boldsymbol{f} \cdot \boldsymbol{v}^h = \int_T f_1 v_1^h + \int_T f_2 v_2^h.$$

The procedure used for both integrals on the right will be similar:

$$f_{Ti} = \frac{1}{3}(f_i(\mathbf{p}_1) + f_i(\mathbf{p}_2) + f_i(\mathbf{p}_3)), \quad i = 1, 2,$$
$$\mathbf{b}_{iT} = \frac{1}{3}|T| f_{Ti}[1, 1, 1]^\mathsf{T}, \qquad i = 1, 2,$$
$$b_{ibT} = \frac{9}{20}|T| f_{Ti}, \qquad i = 1, 2,$$

where $\mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{p}_3$ are again the vertices of the triangle T. The local vector of the right side has the form:

$$\mathbf{b}_T = [\mathbf{b}_{1T}^\mathsf{T}, b_{1bT}, \mathbf{b}_{2T}^\mathsf{T}, b_{2bT}]^\mathsf{T}.$$

Assembly of the linear system

Let us denote:

$$\tilde{\mathbb{A}}_T := \mathbb{M}_T + \mathbb{R}_T + \mathbb{C}_T = \left[\begin{array}{cc|cc} \mathbf{A}_T & \mathbf{z}_{uT} & 0 & 0 \\ \mathbf{z}_{lT}^\mathsf{T} & \omega_T & 0 & 0 \\ \hline 0 & 0 & \mathbf{A}_T & \mathbf{z}_{uT} \\ 0 & 0 & \mathbf{z}_{lT}^\mathsf{T} & \omega_T \end{array}\right],$$

where $\mathbf{A}_T = \mathbf{M}_T + \mathbf{R}_T + \mathbf{C}_T$, $\omega_T = \omega_M$, $\mathbf{z}_{uT} = \mathbf{m}_T + \mathbf{c}_{uT}$ and $\mathbf{z}_{lT} = \mathbf{m}_T + \mathbf{c}_{lT}$. To make the notation clearer, we will omit the lower designation T, however, we still work with the local

object on the triangle T. The local system of linear equations has the form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{z}_u & 0 & 0 & \mathbf{B}_1^\mathsf{T} \\ \mathbf{z}_l^\mathsf{T} & \omega & 0 & 0 & \mathbf{B}_{b1}^\mathsf{T} \\ 0 & 0 & \mathbf{A} & \mathbf{z}_u & \mathbf{B}_2^\mathsf{T} \\ 0 & 0 & \mathbf{z}_l^\mathsf{T} & \omega & \mathbf{B}_{2b}^\mathsf{T} \\ \mathbf{B}_1 & \mathbf{B}_{1b} & \mathbf{B}_2 & \mathbf{B}_{2b} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_{1b} \\ \mathbf{u}_2 \\ \mathbf{u}_{2b} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_{1b} \\ \mathbf{b}_2 \\ \mathbf{b}_{2b} \\ 0 \end{bmatrix},$$

and this the system we permuted as follows:

$$\begin{bmatrix} \mathbf{A} & 0 & \mathbf{z}_u & 0 & \mathbf{B}_1^\mathsf{T} \\ 0 & \mathbf{A} & 0 & \mathbf{z}_u & \mathbf{B}_2^\mathsf{T} \\ \mathbf{z}_l^\mathsf{T} & 0 & \omega & 0 & \mathbf{B}_{1b}^\mathsf{T} \\ 0 & \mathbf{z}_l^\mathsf{T} & 0 & \omega & \mathbf{B}_{2b}^\mathsf{T} \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_{1b} & \mathbf{B}_{2b} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_{1b} \\ \mathbf{u}_{2b} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_{1b} \\ \mathbf{b}_{2b} \\ 0 \end{bmatrix}. \tag{B.1}$$

If we express the bubble component $\mathbf{u}_{1b}$, and $\mathbf{u}_{2b}$ from the third and fourth equations from (B.1) as follows:

$$\mathbf{u}_{1b} = \omega^{-1}(\mathbf{b}_{1b} - \mathbf{z}_l^\mathsf{T}\mathbf{u}_1 - \mathbf{B}_{1b}^\mathsf{T}\mathbf{p}), \tag{B.2}$$

$$\mathbf{u}_{2b} = \omega^{-1}(\mathbf{b}_{2b} - \mathbf{z}_l^\mathsf{T}\mathbf{u}_2 - \mathbf{B}_{2b}^\mathsf{T}\mathbf{p}), \tag{B.3}$$

and substitute (B.2)-(B.3) into the first, second and fifth equations in (B.1) and rearrange them, we get the local linear system:

$$\begin{bmatrix} \mathbb{A} & \mathbb{B}_u^\mathsf{T} \\ \mathbb{B}_l & -\mathbb{E} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbb{b} \\ \mathbb{c} \end{bmatrix}, \tag{B.4}$$

where

$$\begin{aligned} \mathbb{A} &= \begin{bmatrix} \mathbf{A} - \omega^{-1}\mathbf{z}_u\mathbf{z}_l^\mathsf{T} & 0 \\ 0 & \mathbf{A} - \omega^{-1}\mathbf{z}_u\mathbf{z}_l^\mathsf{T} \end{bmatrix}, \\ \mathbb{B}_u &= \begin{bmatrix} \mathbf{B}_1 - \omega^{-1}\mathbf{z}_u\mathbf{B}_{1b} & \mathbf{B}_2 - \omega^{-1}\mathbf{z}_u\mathbf{B}_{2b} \end{bmatrix}, \\ \mathbb{B}_l &= \begin{bmatrix} \mathbf{B}_1 - \omega^{-1}\mathbf{z}_l^\mathsf{T}\mathbf{B}_{1b} & \mathbf{B}_2 - \omega^{-1}\mathbf{z}_l^\mathsf{T}\mathbf{B}_{2b} \end{bmatrix}, \\ \mathbb{E} &= \omega^{-1}(\mathbf{B}_1\mathbf{B}_{1b}^\mathsf{T} + \mathbf{B}_2\mathbf{B}_{2b}^\mathsf{T}), \\ \mathbb{b} &= \begin{bmatrix} \mathbf{b}_1 - \omega^{-1}\mathbf{z}_u\mathbf{b}_{1b} \\ \mathbf{b}_2 - \omega^{-1}\mathbf{z}_u\mathbf{b}_{2b} \end{bmatrix}, \\ \mathbb{c} &= -\omega^{-1}(\mathbf{b}_{1b}\mathbf{b}_{1b} + \mathbf{b}_{2b}\mathbf{b}_{2b}) \end{aligned}$$

and $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)^\mathsf{T}$.

Assembly codes for two dimension

We will now show the non-vectorized and vectorized code in Listing 1 and 2 respectively. The vectorized code 2 was optimized as much as possible.

```
function [A,Bl,Bu,E,b,c]=assembly2DP1bP1C(p,t,alpha,nu,f1,f2,w)
% Assembly the linear system
%|A  Bu'||u|=|b|
%|Bl -E ||p|=|c|
% Matrices are permuted by variable t2 to format as follows:
% [ux_1,uy_1,ux_2,uy_2,...,ux_n,uy_n,p_1,p_2,...,p_n]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   np=size(p,1); nt=size(t,1);
   Ah=sparse(np,np);  Z=sparse(np,np);
   Bu1=sparse(np,np); Bu2=sparse(np,np);
   Bl1=sparse(np,np); Bl2=sparse(np,np);
   E=sparse(np,np);
   b1=zeros(np,1); b2=zeros(np,1); c=zeros(np,1);
   w1=w(1:2:2*np-1);w2=w(2:2:2*np);
   Mel=(1/12)*[2 1 1; 1 2 1; 1 1 2];
   for k=1:nt
      idt=t(k,:);
      % triangle area
      x21=p(t(k,2),1)-p(t(k,1),1); y12=p(t(k,1),2)-p(t(k,2),2);
      x32=p(t(k,3),1)-p(t(k,2),1); y23=p(t(k,2),2)-p(t(k,3),2);
      x13=p(t(k,1),1)-p(t(k,3),1); y31=p(t(k,3),2)-p(t(k,1),2);
      tarea=(x21*y31-x13*y12)/2;
      xt=[x32;x13;x21]; yt=[y23;y31;y12];
      % Sub calcualtions
      w1t=(w1(idt(1))+w1(idt(2))+w1(idt(3)))/3;
      w2t=(w2(idt(1))+w2(idt(2))+w2(idt(3)))/3;
      mt=(3/20)*alpha*tarea*[1;1;1];
      omega_M=(81/280)*alpha*tarea;
      omega_R=(81/40)*nu*(yt(1)^2+xt(1)^2-yt(2)*yt(3)-xt(2)*xt(3))/tarea;
      omega=omega_M+omega_R;
      cut=(9/40)*(w1t*yt+w2t*xt); comega=[1,2,2;2,1,2;2,2,1];
      zu=mt+comega*cut;
      zl=mt+(9/40)*(w1t*yt+w2t*xt);
      f1t=(f1(idt(1))+f1(idt(2))+f1(idt(3)))/3;
      f2t=(f2(idt(1))+f2(idt(2))+f2(idt(3)))/3;
      f1b=(9/20)*f1t*tarea; f2b=(9/20)*f2t*tarea;
      % Assembly matrices
      Ah(idt,idt)=Ah(idt,idt)+alpha*tarea*Mel+(nu/4/tarea)*(yt*yt'+xt*xt')...
                              +[1;1;1]*(w1t*yt'+w2t*xt')/6 -zu*zl'/omega;
      Bu1(idt,idt)=Bu1(idt,idt)-[1;1;1]*yt'/6-(9/40)*(yt*zu')/omega;
      Bu2(idt,idt)=Bu2(idt,idt)-[1;1;1]*xt'/6-(9/40)*(xt*zu')/omega;
      Bl1(idt,idt)=Bl1(idt,idt)-[1;1;1]*yt'/6-(9/40)*(yt*zl')/omega;
      Bl2(idt,idt)=Bl2(idt,idt)-[1;1;1]*xt'/6-(9/40)*(xt*zl')/omega;
      E(idt,idt)=E(idt,idt)-(81/1600)*(yt*yt'+xt*xt')/omega;
      % Assembly vectors
      b1(idt)=b1(idt)+f1t*tarea*[1;1;1]/3-(zu*f1b)/omega;
      b2(idt)=b2(idt)+f2t*tarea*[1;1;1]/3-(zu*f2b)/omega;
      c(idt)=c(idt)-(9/40)*(yt*f1b+xt*f2b)/omega;
   end
   t2(1:2:2*np-1)=1:np; t2(2:2:2*np)=np+[1:np];
   A=[Ah  Z
      Z  Ah];
   A=A(t2,t2);
   Bu=[Bu1 Bu2]; Bu=Bu(:,t2);
   Bl=[Bl1 Bl2]; Bl=Bl(:,t2);
   b=[b1;b2]; b=b(t2);
end
```

Listing 1: Non-vectorized 2D assembly function

```
function [A,Bl,Bu,E,b,c]=assembly2DP1bP1C_vec(p,t,alpha,nu,f1,f2,w)
% Assembly the linear system
%|A  Bu'||u|=|b|
%|Bl -E ||p|=|c|
% Matrices are permuted by variable t2 to format as follows:
% [ux_1,uy_1,ux_2,uy_2,...,ux_n,uy_n,p_1,p_2,...,p_n]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    np=size(p,1);
    Ah=sparse(np,np);  Z=sparse(np,np);
    Bu1=sparse(np,np); Bu2=sparse(np,np);
    Bl1=sparse(np,np); Bl2=sparse(np,np);
    E=sparse(np,np);
    b1=sparse(np,1); b2=sparse(np,1); c=sparse(np,1);
    % triangles area
    x21=p(t(:,2),1)-p(t(:,1),1); y12=p(t(:,1),2)-p(t(:,2),2);
    x32=p(t(:,3),1)-p(t(:,2),1); y23=p(t(:,2),2)-p(t(:,3),2);
    x13=p(t(:,1),1)-p(t(:,3),1); y31=p(t(:,3),2)-p(t(:,1),2);
    xt=[x32 x13 x21]; yt=[y23 y31 y12];
    tarea=(x21.*y31-x13.*y12)/2;
    % Sub calculations
    w1=w(1:2:2*np-1); w2=w(2:2:2*np);
    w1t=(w1(t(:,1))+w1(t(:,2))+w1(t(:,3)))/3;
    w2t=(w2(t(:,1))+w2(t(:,2))+w2(t(:,3)))/3;
    w1yw2x=[w1t.*yt(:,1)+w2t.*xt(:,1),
            w1t.*yt(:,2)+w2t.*xt(:,2),
            w1t.*yt(:,3)+w2t.*xt(:,3)];
    mt=(3/20)*alpha*tarea;
    omega_M=(81/280)*alpha*tarea;
    omega_R=(81/40)*nu*(y23.^2+x32.^2-y31.*y12-x13.*x21)./tarea;
    omega=omega_M+omega_R;
    zu(:,1)=mt+(27/120)*(1*w1yw2x(:,1)+2*w1yw2x(:,2)+2*w1yw2x(:,3));
    zu(:,2)=mt+(27/120)*(2*w1yw2x(:,1)+1*w1yw2x(:,2)+2*w1yw2x(:,3));
    zu(:,3)=mt+(27/120)*(2*w1yw2x(:,1)+2*w1yw2x(:,2)+1*w1yw2x(:,3));
    zl=mt+(9/40)*(w1t.*yt+w2t.*xt);
    f1t=(f1(t(:,1))+f1(t(:,2))+f1(t(:,3)))/3;
    f2t=(f2(t(:,1))+f2(t(:,2))+f2(t(:,3)))/3;
    f1b=(9/20)*tarea.*f1t;
    f2b=(9/20)*tarea.*f2t;
    % Mass matrix
    for i=1:3
        for j=1:i
            Ah=Ah+sparse(t(:,i),t(:,j),alpha*tarea/12,np,np);
        end
    end
    Ah=Ah+Ah.';
    % Sub compute indemendent of i,j
    cal=(1/6)*w1yw2x;
    f1tt=(1/3)*f1t.*tarea;
    f2tt=(1/3)*f2t.*tarea;
    f1bo=f1b./omega;
    f2bo=f2b./omega;
    fbb1o=(9/40)*f1b./omega;
    fbb2o=(9/40)*f2b./omega;
    for i=1:3
        % Assembly vectors
        b1=b1+sparse(t(:,i),1,f1tt-zu(:,i).*f1bo,np,1);
        b2=b2+sparse(t(:,i),1,f2tt-zu(:,i).*f2bo,np,1);
        c=c-sparse(t(:,i),1,yt(:,i).*fbb1o+xt(:,i).*fbb2o,np,1);
        % Sub calcualtion for matrices independent of i
        nuxtt=(nu/4)*xt(:,i)./tarea;
        nuytt=(nu/4)*yt(:,i)./tarea;
        zuom=zu(:,i)./omega;
        nuxtom=(9/40)*xt(:,i)./omega;
        nuytom=(9/40)*yt(:,i)./omega;
```

```
        extom=(81/1600)*xt(:,i)./omega;
        eytom=(81/1600)*yt(:,i)./omega;
        for j=1:3
            % Assembly matrices
            Ah=Ah+sparse(t(:,i),t(:,j),nuytt.*yt(:,j)+nuxtt.*xt(:,j)...
                                        +cal(:,j)-zl(:,j).*zuom,np,np);
            Bu1=Bu1-sparse(t(:,i),t(:,j),yt(:,j)/6+zu(:,j).*nuytom,np,np);
            Bu2=Bu2-sparse(t(:,i),t(:,j),xt(:,j)/6+zu(:,j).*nuxtom,np,np);
            Bl1=Bl1-sparse(t(:,i),t(:,j),yt(:,j)/6+zl(:,j).*nuytom,np,np);
            Bl2=Bl2-sparse(t(:,i),t(:,j),xt(:,j)/6+zl(:,j).*nuxtom,np,np);
            E=E-sparse(t(:,i),t(:,j),eytom.*yt(:,j)+extom.*xt(:,j),np,np);
        end
    end
    t2(1:2:2*np-1)=1:np; t2(2:2:2*np)=np+[1:np];
    A=[Ah  Z
        Z  Ah];
    A=A(t2,t2);
    Bu=[Bu1,Bu2]; Bu=Bu(:,t2);
    Bl=[Bl1,Bl2]; Bl=Bl(:,t2);
    b=[full(b1);full(b2)]; b=b(t2);
    c=full(c)
end
```

Listing 2: Vectorized 2D assembly function

C Assembly function in three dimensions

Let us have tetrahedrons discretization $\mathcal{T}_h$ of the domain $\Omega \subset \mathbb{R}^3$. On the tetrahedron $T \in \mathcal{T}_h$ with vertices $\mathbf{p}_i = [x_i, y_i, z_i]^\mathsf{T}$, $i = 1, 2, 3, 4$ we will have four nonzero linear basis function $\phi_1(\boldsymbol{x})$, $\phi_2(\boldsymbol{x})$, $\phi_3(\boldsymbol{x})$, $\phi_4(\boldsymbol{x}) \in P_1(T)$, $\boldsymbol{x} = [x, y, z]^\mathsf{T} \in T$ defined by conditions: $\phi_i(\mathbf{p}_j) = \delta_{ij}$, $i, j = 1, 2, 3, 4$ and also the bubble function defined on T as follows:

$$\phi_b(\boldsymbol{x}) = 4^4 \cdot \phi_1(\boldsymbol{x}) \cdot \phi_2(\boldsymbol{x}) \cdot \phi_3(\boldsymbol{x}) \cdot \phi_4(\boldsymbol{x}), \quad \boldsymbol{x} \in T.$$

For simplicity, we will denote $\phi_5 = \phi_b$. We will aproximate the components of the velocity vector $\boldsymbol{u}^h = (u_1^h, u_2^h, u_3^h)^\mathsf{T}$ and the pressure p^h on T as follows:

$$u_k^h = \sum_{j=1}^{5} u_{kj}\phi_j, \quad k = 1, 2, 3,$$
$$p^h = \sum_{j=1}^{4} p_j \phi_j.$$

Mass matrix

As in the two dimensions, we first derive the mass matrix from the integral:

$$\alpha \int_T \boldsymbol{u}^h \cdot \boldsymbol{v}^h = \alpha \int_T u_1^h v_1^h + \alpha \int_T u_2^h v_2^h + \alpha \int_T u_3^h v_3^h,$$

where $\boldsymbol{v}^h = (v_1^h, v_2^h, v_3^h)^\mathsf{T}$. For $\boldsymbol{v}^h = (\phi_i, 0, 0)^\mathsf{T}$, $\boldsymbol{v}^h = (0, \phi_i, 0)^\mathsf{T}$, and $\boldsymbol{v}^h = (0, \phi_i)^\mathsf{T}$ we get:

$$\sum_{j=1}^{5} u_{1j}\alpha \int_T \phi_j\phi_i, \quad i = 1, \ldots, 5,$$
$$\sum_{j=1}^{5} u_{2j}\alpha \int_T \phi_j\phi_i, \quad i = 1, \ldots, 5,$$
$$\sum_{j=1}^{5} u_{3j}\alpha \int_T \phi_j\phi_i, \quad i = 1, \ldots, 5,$$

respectively. The local mass matrix $\mathbb{M}_T \in \mathbb{R}^{15\times 15}$ therefore form as follow:

$$\mathbb{M}_T = \left[\begin{array}{cc|cc|cc} \mathbf{M}_T & \mathbf{m}_T & 0 & 0 & 0 & 0 \\ \mathbf{m}_T^\mathsf{T} & \omega_M & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \mathbf{M}_T & \mathbf{m}_T & 0 & 0 \\ 0 & 0 & \mathbf{m}_T^\mathsf{T} & \omega_M & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \mathbf{M}_T & \mathbf{m}_T \\ 0 & 0 & 0 & 0 & \mathbf{m}_T^\mathsf{T} & \omega_M \end{array}\right],$$

where $\mathbf{M}_T,\ \mathbf{m}_T,\ \omega_M$ are also derived in [8] and read as follows:

$$\mathbf{M}_T = \frac{\alpha|T|}{20}\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}, \quad \mathbf{m}_T = \frac{8\alpha|T|}{105}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \omega_M = \frac{621\alpha|T|}{3940}.$$

Stiffness matrix

Next, let us derive the local stiffness matrix $\mathbb{R}_T \in \mathbb{R}^{15\times 15}$. The respective integral has the following form:

$$\nu\int_T \nabla \boldsymbol{u}^h : \nabla \boldsymbol{v}^h = \nu\int_T \nabla \boldsymbol{u}_1^h \cdot \nabla \boldsymbol{v}_1^h + \nu\int_T \nabla \boldsymbol{u}_2^h \cdot \nabla \boldsymbol{v}_2^h + \nu\int_T \nabla \boldsymbol{u}_3^h \cdot \nabla \boldsymbol{v}_3^h =$$
$$= \nu\int_T u_{1x}^h v_{1x}^h + u_{1y}^h v_{1y}^h + u_{1z}^h v_{1z}^h + u_{2x}^h v_{2x}^h + u_{2y}^h v_{2y}^h + u_{2z}^h v_{2z}^h + u_{3x}^h v_{3x}^h + u_{3y}^h v_{3y}^h + u_{3z}^h v_{3z}^h.$$

For $\boldsymbol{v}^h = (\phi_i, 0, 0)^\mathsf{T}$ we get:

$$\sum_{j=1}^{5} u_{1j}\nu\int_T (\phi_{jx}\phi_{ix} + \phi_{jy}\phi_{iy} + \phi_{jz}\phi_{iz}), \quad \text{for } i = 1, \ldots, 5,$$

for $\boldsymbol{v}^h = (0, \phi_i, 0)^\mathsf{T}$:

$$\sum_{j=1}^{5} u_{2j}\nu\int_T (\phi_{jx}\phi_{ix} + \phi_{jy}\phi_{iy} + \phi_{jz}\phi_{iz}), \quad \text{for } i = 1, \ldots, 5,$$

and for $\boldsymbol{v}^h = (0, 0, \phi_i)^\mathsf{T}$:

$$\sum_{j=1}^{5} u_{3j}\nu\int_T (\phi_{jx}\phi_{ix} + \phi_{jy}\phi_{iy} + \phi_{jz}\phi_{iz}), \quad \text{for } i = 1, \ldots, 5.$$

The local stiffness matrix $\mathbb{R}_T \in \mathbb{R}^{15\times 15}$ have the form as follows:

$$\mathbb{R}_T = \left[\begin{array}{cc|cc|cc} \mathbf{R}_T & \mathbf{r}_T & 0 & 0 & 0 & 0 \\ \mathbf{r}_T^\mathsf{T} & \omega_R & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \mathbf{R}_T & \mathbf{r}_T & 0 & 0 \\ 0 & 0 & \mathbf{r}_T^\mathsf{T} & \omega_R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \mathbf{R}_T & \mathbf{r}_T \\ 0 & 0 & 0 & 0 & \mathbf{r}_T^\mathsf{T} & \omega_R \end{array}\right],$$

where also

$$(\mathbf{R}_T)_{ij} = \nu \int_T (\phi_{jx}\phi_{ix} + \phi_{jy}\phi_{iy} + \phi_{jz}\phi_{iz}), \quad i,j = 1,2,3,4,$$
$$\omega_R = \nu \int_T (\phi_{bx}\phi_{bx} + \phi_{by}\phi_{by} + \phi_{bz}\phi_{bz}),$$
$$(\mathbf{r}_T)_j = \nu \int_T (\phi_{jx}\phi_{bx} + \phi_{jy}\phi_{by} + \phi_{jz}\phi_{bz}) = 0, \quad j = 1,2,3,4,$$

Let $\mathbf{p}_1 = [x_1, y_1, z_1]^\mathsf{T}$, $\mathbf{p}_2 = [x_2, y_2, z_2]^\mathsf{T}$, $\mathbf{p}_3 = [x_3, y_3, z_3]^\mathsf{T}$, and $\mathbf{p}_4 = [x_4, y_4, z_4]^\mathsf{T}$ are the vertices of the tetrahedron T. To simplify our presentation, we will take over the marking from [8], which is described in detail here

$$\boldsymbol{x}_T = \begin{bmatrix} y_{42}\, z_{32} - y_{32}\, z_{42} \\ y_{31}\, z_{41} - y_{41}\, z_{31} \\ y_{41}\, z_{21} - y_{21}\, z_{41} \\ y_{21}\, z_{31} - y_{31}\, z_{21} \end{bmatrix}, \quad \boldsymbol{y}_T = \begin{bmatrix} x_{32}\, z_{42} - x_{42}\, z_{32} \\ z_{31}\, x_{41} - z_{41}\, x_{31} \\ z_{41}\, x_{21} - z_{21}\, x_{41} \\ z_{21}\, x_{31} - z_{31}\, x_{21} \end{bmatrix}, \quad \boldsymbol{z}_T = \begin{bmatrix} x_{42}\, y_{32} - x_{32}\, y_{42} \\ x_{31}\, y_{41} - x_{41}\, y_{31} \\ x_{41}\, y_{21} - x_{21}\, y_{41} \\ x_{21}\, y_{31} - x_{31}\, y_{21} \end{bmatrix}.$$

Then is proved in [8] also:

$$\left[\frac{\partial\phi_1}{\partial x}, \frac{\partial\phi_2}{\partial x}, \frac{\partial\phi_3}{\partial x}, \frac{\partial\phi_4}{\partial x}\right]^\mathsf{T} = \frac{1}{6|T|}\boldsymbol{x}_T,$$
$$\left[\frac{\partial\phi_1}{\partial y}, \frac{\partial\phi_2}{\partial y}, \frac{\partial\phi_3}{\partial y}, \frac{\partial\phi_4}{\partial y}\right]^\mathsf{T} = \frac{1}{6|T|}\boldsymbol{y}_T,$$
$$\left[\frac{\partial\phi_1}{\partial z}, \frac{\partial\phi_2}{\partial z}, \frac{\partial\phi_3}{\partial z}, \frac{\partial\phi_4}{\partial z}\right]^\mathsf{T} = \frac{1}{6|T|}\boldsymbol{z}_T,$$

where $6|T|$ is the volume of the tetrahedron T.
Finally, we get:

$$\mathbf{R}_T = \frac{\nu}{36|T|}(\boldsymbol{x}_T\boldsymbol{x}_T^\mathsf{T} + \boldsymbol{y}_T\boldsymbol{y}_T^\mathsf{T} + \boldsymbol{z}_T\boldsymbol{z}_T^\mathsf{T}),$$

$$\omega_R = \frac{759}{3152|T|}(x_{T1}^2 + y_{T1}^2 + z_{T1}^2 - x_{T2}x_{T3} - y_{T2}y_{T3} - z_{T2}z_{T3} - x_{T2}x_{T4} -$$
$$- y_{T2}y_{T4} - z_{T2}z_{T4} - x_{T3}x_{T4} - y_{T3}y_{T4} - z_{T3}z_{T4}).$$

Convective matrix

The local convective matrix we obtain from the integrals for the convective term (A.4) for $d = 3$ reads as follows:

$$\int_T (\boldsymbol{w}\cdot\nabla\boldsymbol{u}^h)\cdot\boldsymbol{v}^h = \int_T \left(\begin{bmatrix} w_1 u_{1x}^h + w_2 u_{1y}^h + w_3 u_{1z}^h \\ w_1 u_{2x}^h + w_2 u_{2y}^h + w_3 u_{2z}^h \\ w_1 u_{3x}^h + w_2 u_{3y}^h + w_3 u_{3z}^h \end{bmatrix} \cdot [v_1^h, v_2^h, v_3^h] \right) =$$

$$= \int_T (w_1 u^h_{1x} v^h_1 + w_2 u^h_{1y} v^h_1 + w_3 u^h_{1z} v^h_1) + \int_T (w_1 u^h_{2x} v^h_2 + w_2 u^h_{2y} v^h_2 + w_3 u^h_{2z} v^h_2) +$$
$$+ \int_T (w_1 u^h_{3x} v^h_3 + w_2 u^h_{3y} v^h_3 + w_3 u^h_{3z} v^h_3).$$

For $\boldsymbol{v}^h = (\phi_i, 0, 0)^\mathsf{T}$ we get:

$$\sum_{j=1}^{5} u_{1j} \int_T (w_1 \phi_{jx} \phi_i + w_2 \phi_{jy} \phi_i + w_3 \phi_{jz} \phi_i), \quad \text{for } i = 1, \ldots, 5,$$

for $\boldsymbol{v}^h = (0, \phi_i, 0)^\mathsf{T}$:

$$\sum_{j=1}^{5} u_{2j} \int_T (w_1 \phi_{jx} \phi_i + w_2 \phi_{jy} \phi_i + w_3 \phi_{jz} \phi_i), \quad \text{for } i = 1, \ldots, 5,$$

and for $\boldsymbol{v}^h = (0, 0, \phi_i)^\mathsf{T}$:

$$\sum_{j=1}^{5} u_{3j} \int_T (w_1 \phi_{jx} \phi_i + w_2 \phi_{jy} \phi_i + w_3 \phi_{jz} \phi_i), \quad \text{for } i = 1, \ldots, 5.$$

The local convective matrix $\mathbb{C}_T \in \mathbb{R}^{15\times 15}$ have the form as follows:

$$\mathbb{C}_T = \left[\begin{array}{cc|cc|cc} \mathbf{C}_T & \mathbf{c}_{uT} & 0 & 0 & 0 & 0 \\ \mathbf{c}_{lT}^\mathsf{T} & \omega_C & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \mathbf{C}_T & \mathbf{c}_{uT} & 0 & 0 \\ 0 & 0 & \mathbf{c}_{lT}^\mathsf{T} & \omega_C & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \mathbf{C}_T & \mathbf{c}_{uT} \\ 0 & 0 & 0 & 0 & \mathbf{c}_{lT}^\mathsf{T} & \omega_C \end{array}\right],$$

In [8] is proved that

$$\int_T \phi_i = \frac{6|T|}{24}, \quad i = 1, \ldots, 4.$$

Let us denote the approximation of $\boldsymbol{w}$ as follows:

$$w_k^{(T)} = \frac{1}{4}(w_{k,1} + w_{k,2} + w_{k,3} + w_{k,4}), \quad k = 1, 2, 3,$$

where $w_{k,i}$ represents the value of the each *i-th* vertex of the tetrahedron T, then for $i, j = 1, \ldots, 4$,

$$(\mathbf{C}_T)_{ij} := w_1^{(T)} \phi_{jx} \int_T \phi_i + w_2^{(T)} \phi_{jy} \int_T \phi_i + w_3^{(T)} \phi_{jz} \int_T \phi_i =$$
$$= w_1^{(T)} x_{Tj} \frac{1}{6|T|} \frac{6|T|}{24} + w_2^{(T)} y_{Tj} \frac{1}{6|T|} \frac{6|T|}{24} + w_3^{(T)} z_{Tj} \frac{1}{6|T|} \frac{6|T|}{24}.$$

We get:

$$\mathbf{C}_T = w_1^{(T)} \frac{1}{24} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{x}_T^{\mathsf{T}} + w_2^{(T)} \frac{1}{24} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{y}_T^{\mathsf{T}} + w_3^{(T)} \frac{1}{24} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{z}_T^{\mathsf{T}}.$$

The bubble component ω_C has form as follows:

$$\begin{aligned}
\omega_C :=& \int_T w_1^{(T)} \phi_{bx} \phi_b + w_2^{(T)} \phi_{by} \phi_b + w_3^{(T)} \phi_{bz} \phi_b = \\
=& 256^2 w_1^{(T)} \left[\phi_{1x} \int_T \phi_1 \phi_2^2 \phi_3^2 \phi_4^2 + \phi_{2x} \int_T \phi_1^2 \phi_2 \phi_3^2 \phi_4^2 + \phi_{3x} \int_T \phi_1^2 \phi_2^2 \phi_3 \phi_4^2 + \phi_{4x} \int_T \phi_1^2 \phi_2^2 \phi_3^2 \phi_4 \right] + \\
&+ 256^2 w_2^{(T)} \left[\phi_{1y} \int_T \phi_1 \phi_2^2 \phi_3^2 \phi_4^2 + \phi_{2y} \int_T \phi_1^2 \phi_2 \phi_3^2 \phi_4^2 + \phi_{3y} \int_T \phi_1^2 \phi_2^2 \phi_3 \phi_4^2 + \phi_{4y} \int_T \phi_1^2 \phi_2^2 \phi_3^2 \phi_4 \right] + \\
&+ 256^2 w_3^{(T)} \left[\phi_{1z} \int_T \phi_1 \phi_2^2 \phi_3^2 \phi_4^2 + \phi_{2z} \int_T \phi_1^2 \phi_2 \phi_3^2 \phi_4^2 + \phi_{3z} \int_T \phi_1^2 \phi_2^2 \phi_3 \phi_4^2 + \phi_{4z} \int_T \phi_1^2 \phi_2^2 \phi_3^2 \phi_4 \right],
\end{aligned}$$

where all subintegral are equal as follows:

$$\int_T \phi_1^i \phi_2^j \phi_3^k \phi_4^l = \frac{6|T|}{453600}, \quad i,j,k,l = 1,2 \ \wedge \ i+j+k+l = 7.$$

Then we can write that

$$\omega_C = \frac{318}{2201} w_1^{(T)} [1,1,1,1] \cdot \boldsymbol{x}_T + \frac{318}{2201} w_2^{(T)} [1,1,1,1] \cdot \boldsymbol{y}_T + \frac{318}{2201} w_3^{(T)} [1,1,1,1] \cdot \boldsymbol{z}_T = 0.$$

The sum of the components $x_{T1} + x_{T2} + x_{T3} + x_{T4} = 0$ as well as sums of $\mathbf{y}_T$ and $\mathbf{z}_T$.
Next we describe $\mathbf{c}_{uT}$:

$$(\mathbf{c}_{uT})_i = \int_T w_1 \phi_{bx} \phi_i + w_2 \phi_{by} \phi_i + w_3 \phi_{bz} \phi_i, \quad i = 1,2,3,4.$$

First we calculate the integrals:

$$\begin{aligned}
\int_T \phi_1 \phi_2 \phi_3 \phi_4 &= \frac{6|T|}{5040}, \\
\int_T \phi_1^2 \phi_3 \phi_4 = \int_T \phi_1^2 \phi_2 \phi_4 &= \int_T \phi_1^2 \phi_2 \phi_3 = \frac{6|T|}{2520}, \\
\int_T \phi_2^2 \phi_3 \phi_4 = \int_T \phi_1 \phi_2^2 \phi_4 &= \int_T \phi_1 \phi_2^2 \phi_3 = \frac{6|T|}{2520}, \\
\int_T \phi_2 \phi_3^2 \phi_4 = \int_T \phi_1 \phi_3^2 \phi_4 &= \int_T \phi_1 \phi_2 \phi_3^2 = \frac{6|T|}{2520}, \\
\int_T \phi_2 \phi_3 \phi_4^2 = \int_T \phi_1 \phi_3 \phi_4^2 &= \int_T \phi_1 \phi_2 \phi_4^2 = \frac{6|T|}{2520},
\end{aligned}$$

Then for $i = 1$:

$$
\begin{aligned}
(\mathbf{c}_{uT})_1 &= \int_T w_1\phi_{bx}\phi_1 + w_2\phi_{by}\phi_1 + w_3\phi_{bz}\phi_1 = \\
&= 256w_1^{(T)}\left[\phi_{1x}\int_T \phi_1\phi_2\phi_3\phi_4 + \phi_{2x}\int_T \phi_1^2\phi_3\phi_4 + \phi_{3x}\int_T \phi_1^2\phi_2\phi_4 + \phi_{4x}\int_T \phi_1^2\phi_2\phi_3\right] + \\
&+ 256w_2^{(T)}\left[\phi_{1y}\int_T \phi_1\phi_2\phi_3\phi_4 + \phi_{2y}\int_T \phi_1^2\phi_3\phi_4 + \phi_{3y}\int_T \phi_1^2\phi_2\phi_4 + \phi_{4y}\int_T \phi_1^2\phi_2\phi_3\right] + \\
&+ 256w_3^{(T)}\left[\phi_{1z}\int_T \phi_1\phi_2\phi_3\phi_4 + \phi_{2z}\int_T \phi_1^2\phi_3\phi_4 + \phi_{3z}\int_T \phi_1^2\phi_2\phi_4 + \phi_{4z}\int_T \phi_1^2\phi_2\phi_3\right] = \\
&= \frac{256}{5040}w_1^{(T)}\left[x_{T1} + 2x_{T2} + 2x_{T3} + 2x_{T4}\right] + \frac{256}{5040}w_2^{(T)}\left[y_{T1} + 2y_{T2} + 2y_{T3} + 2y_{T4}\right] + \\
&+ \frac{256}{5040}w_3^{(T)}\left[z_{T1} + 2z_{T2} + 2z_{T3} + 2z_{T4}\right] = \\
&= \frac{16}{315}w_1^{(T)}[1,2,2,2]\cdot\boldsymbol{x}_T + \frac{16}{315}w_2^{(T)}[1,2,2,2]\cdot\boldsymbol{y}_T + \frac{16}{315}w_3^{(T)}[1,2,2,2]\cdot\boldsymbol{z}_T = \\
&= \frac{16}{315}[1,2,2,2]\cdot(w_1^{(T)}\boldsymbol{x}_T + w_2^{(T)}\boldsymbol{y}_T + w_3^{(T)}\boldsymbol{z}_T).
\end{aligned}
$$

Let us denote $\boldsymbol{c}_1 = [1,2,2,2]$. For $i = 2,3,4$ we calculate that $\boldsymbol{c}_2 = [2,1,2,2]$, $\boldsymbol{c}_3 = [2,2,1,2]$, and $\boldsymbol{c}_4 = [2,2,2,1]$, then we can write as follows:

$$
(\mathbf{c}_{uT})_i = \frac{16}{315}\boldsymbol{c}_i\cdot(w_1\boldsymbol{x}_T + w_2\boldsymbol{y}_T + w_3\boldsymbol{z}_T) \quad i = 1,2,3,4,
$$

and finally

$$
\mathbf{c}_{uT} = \frac{16}{315}\begin{bmatrix} 1 & 2 & 2 & 2 \\ 2 & 1 & 2 & 2 \\ 2 & 2 & 1 & 2 \\ 2 & 2 & 2 & 1 \end{bmatrix}(w_1\boldsymbol{x}_T + w_2\boldsymbol{y}_T + w_3\boldsymbol{z}_T).
$$

As the last member we describre $\mathbf{c}_{lT}$, for $j = 1,2,3,4$:

$$
\begin{aligned}
(\mathbf{c}_{lT})_j &= \int_T w_1\phi_{jx}\phi_b + w_2\phi_{jy}\phi_b + w_3\phi_{jz}\phi_b = \\
&= 256w_1^{(T)}\phi_{jx}\int_T \phi_1\phi_2\phi_3\phi_4 + 256w_2^{(T)}\phi_{jy}\int_T \phi_1\phi_2\phi_3\phi_4 + 256w_3^{(T)}\phi_{jz}\int_T \phi_1\phi_2\phi_3\phi_4 = \\
&= 256w_1^{(T)}\frac{1}{6|T|}x_{Tj}\frac{6|T|}{5040} + 256w_2^{(T)}\frac{1}{6|T|}y_{Tj}\frac{6|T|}{5040} + 256w_3^{(T)}\frac{1}{6|T|}z_{Tj}\frac{6|T|}{5040}.
\end{aligned}
$$

We get:

$$
\mathbf{c}_{lT} = \frac{16}{315}\left(w_1^{(T)}\mathbf{x}_T^{\mathsf{T}} + w_2^{(T)}\mathbf{y}_T^{\mathsf{T}} + w_3^{(T)}\mathbf{z}_T^{\mathsf{T}}\right).
$$

Divergence and gradient matrix

We know from the two-dimensional problem that the local gradient matrix is divergent by transposition and therefore does not need to be assembled. The local divergence matrix is given

by the approximation of the integral in the second equation in problem (P). We come out of the expression:

$$-\int_T q^h(u_{1x}^h + u_{2y}^h + u_{3z}^h),$$

where we substitute the approximations $u_1^h,\ u_2^h,\ u_3^h$, and for q^h we gradually choose all the basis functions used in the pressure approximation. We will get:

$$\sum_{j=1}^{5} u_{1j}\left(-\int_T \phi_i\phi_{jx}\right) + \sum_{j=1}^{5} u_{2j}\left(-\int_T \phi_i\phi_{jy}\right) + \sum_{j=1}^{5} u_{3j}\left(-\int_T \phi_i\phi_{jy}\right), \quad i = 1,\dots,4.$$

The local divergence matrix $\mathbb{B}_T \in \mathbb{R}^{4\times 15}$ should be divided into parts reads as follows:

$$\mathbb{B}_T = [\mathbf{B}_{1T}, \mathbf{B}_{1bT}, \mathbf{B}_{2T}, \mathbf{B}_{2bT}, \mathbf{B}_{3T}, \mathbf{B}_{3bT}],$$

where

$$\begin{aligned}
\mathbf{B}_{1T} &= \frac{-\mathrm{sgn}|\mathbf{X}|}{24}[\boldsymbol{x}_T, \boldsymbol{x}_T, \boldsymbol{x}_T, \boldsymbol{x}_T]^\mathsf{T}, & \mathbf{B}_{1bT} &= \frac{16}{315}\mathrm{sgn}|\mathbf{X}|\boldsymbol{x}_T,\\
\mathbf{B}_{2T} &= \frac{-\mathrm{sgn}|\mathbf{X}|}{24}[\boldsymbol{y}_T, \boldsymbol{y}_T, \boldsymbol{y}_T, \boldsymbol{y}_T]^\mathsf{T}, & \mathbf{B}_{2bT} &= \frac{16}{315}\mathrm{sgn}|\mathbf{X}|\boldsymbol{y}_T,\\
\mathbf{B}_{3T} &= \frac{-\mathrm{sgn}|\mathbf{X}|}{24}[\boldsymbol{z}_T, \boldsymbol{z}_T, \boldsymbol{z}_T, \boldsymbol{z}_T]^\mathsf{T}, & \mathbf{B}_{3bT} &= \frac{16}{315}\mathrm{sgn}|\mathbf{X}|\boldsymbol{z}_T,
\end{aligned}$$

which is proved in [8].

Vector of the right side

The local vector of the right side arises from the approximation of the integral:

$$\int_T \boldsymbol{f}\cdot\boldsymbol{v}^h = \int_T f_1 v_1^h + \int_T f_2 v_2^h + \int_T f_3 v_3^h.$$

The procedure used for all integrals on the right will be similar:

$$\begin{aligned}
f_{iT} &= \frac{1}{4}(f_i(\mathbf{p}_1) + f_i(\mathbf{p}_2) + f_i(\mathbf{p}_3) + f_i(\mathbf{p}_4)), & i &= 1,2,3,\\
\mathbf{b}_{iT} &= \frac{1}{4}|T| f_{iT}[1,1,1,1]^\mathsf{T}, & i &= 1,2,3,\\
b_{ibT} &= \frac{32}{105}|T| f_{iT}, & i &= 1,2,3.
\end{aligned}$$

The local vector of the right side has the form:

$$\mathbf{b}_T = [\mathbf{b}_{1T}^\mathsf{T}, b_{1bT}, \mathbf{b}_{2T}^\mathsf{T}, b_{2bT}, \mathbf{b}_{3T}^\mathsf{T}, b_{3bT}]^\mathsf{T}.$$

Assembly of the linear system

Let us denote:

$$\tilde{\mathbb{A}}_T := \mathbb{M}_T + \mathbb{R}_T + \mathbb{C}_T = \left[\begin{array}{cc|cc|cc} \mathbf{A}_T & \mathbf{z}_{uT} & 0 & 0 & 0 & 0 \\ \mathbf{z}_{lT}^\mathsf{T} & \omega_T & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \mathbf{A}_T & \mathbf{z}_{uT} & 0 & 0 \\ 0 & 0 & \mathbf{z}_{lT}^\mathsf{T} & \omega_T & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \mathbf{A}_T & \mathbf{z}_{uT} \\ 0 & 0 & 0 & 0 & \mathbf{z}_{lT}^\mathsf{T} & \omega_T \end{array}\right],$$

where $\mathbf{A}_T = \mathbf{M}_T + \mathbf{R}_T + \mathbf{C}_T$, $\omega_T = \omega_M$, $\mathbf{z}_{uT} = \mathbf{m}_T + \mathbf{c}_{uT}$ and $\mathbf{z}_{lT} = \mathbf{m}_T + \mathbf{c}_{lT}$. Here, too, we will simplify the notation by omitting the lower designation T, however, we still work with local object on the tetrahedron T. The local system of linear equations has the form:

$$\left[\begin{array}{ccccccc} \mathbf{A} & \mathbf{z}_u & 0 & 0 & 0 & 0 & \mathbf{B}_1^\mathsf{T} \\ \mathbf{z}_l^\mathsf{T} & \omega & 0 & 0 & 0 & 0 & \mathbf{B}_{1b}^\mathsf{T} \\ 0 & 0 & \mathbf{A} & \mathbf{z}_u & 0 & 0 & \mathbf{B}_2^\mathsf{T} \\ 0 & 0 & \mathbf{z}_l^\mathsf{T} & \omega & 0 & 0 & \mathbf{B}_{2b}^\mathsf{T} \\ 0 & 0 & 0 & 0 & \mathbf{A} & \mathbf{z}_u & \mathbf{B}_3^\mathsf{T} \\ 0 & 0 & 0 & 0 & \mathbf{z}_l^\mathsf{T} & \omega & \mathbf{B}_{3b}^\mathsf{T} \\ \mathbf{B}_1 & \mathbf{B}_{1b} & \mathbf{B}_2 & \mathbf{B}_{2b} & \mathbf{B}_3 & \mathbf{B}_{3b} & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{u}_1 \\ \mathbf{u}_{1b} \\ \mathbf{u}_2 \\ \mathbf{u}_{2b} \\ \mathbf{u}_3 \\ \mathbf{u}_{3b} \\ \mathbf{p} \end{array}\right] = \left[\begin{array}{c} \mathbf{b}_1 \\ \mathbf{b}_{1b} \\ \mathbf{b}_2 \\ \mathbf{b}_{2b} \\ \mathbf{b}_3 \\ \mathbf{b}_{3b} \\ 0 \end{array}\right],$$

and this the system we permuted as follows:

$$\left[\begin{array}{ccccccc} \mathbf{A} & 0 & 0 & \mathbf{z}_u & 0 & 0 & \mathbf{B}_1^\mathsf{T} \\ 0 & \mathbf{A} & 0 & 0 & \mathbf{z}_u & 0 & \mathbf{B}_2^\mathsf{T} \\ 0 & 0 & \mathbf{A} & 0 & 0 & \mathbf{z}_u & \mathbf{B}_3^\mathsf{T} \\ \mathbf{z}_l^\mathsf{T} & 0 & 0 & \omega & 0 & 0 & \mathbf{B}_{1b}^\mathsf{T} \\ 0 & \mathbf{z}_l^\mathsf{T} & 0 & 0 & \omega & 0 & \mathbf{B}_{2b}^\mathsf{T} \\ 0 & 0 & \mathbf{z}_l^\mathsf{T} & 0 & 0 & \omega & \mathbf{B}_{3b}^\mathsf{T} \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{B}_{1b} & \mathbf{B}_{2b} & \mathbf{B}_{3b} & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_{1b} \\ \mathbf{u}_{2b} \\ \mathbf{u}_{3b} \\ \mathbf{p} \end{array}\right] = \left[\begin{array}{c} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_{1b} \\ \mathbf{b}_{2b} \\ \mathbf{b}_{3b} \\ 0 \end{array}\right]. \tag{C.1}$$

Now, we express the bubble component $\mathbf{u}_{1b}$, $\mathbf{u}_{2b}$, and $\mathbf{u}_{3b}$ from the fourt to sixth equations from (C.1) as follows:

$$\mathbf{u}_{1b} = \omega^{-1}(\mathbf{b}_{1b} - \mathbf{z}_l^\mathsf{T}\mathbf{u}_1 - \mathbf{B}_{1b}^\mathsf{T}\mathbf{p}), \tag{C.2}$$

$$\mathbf{u}_{2b} = \omega^{-1}(\mathbf{b}_{2b} - \mathbf{z}_l^\mathsf{T}\mathbf{u}_2 - \mathbf{B}_{2b}^\mathsf{T}\mathbf{p}), \tag{C.3}$$

$$\mathbf{u}_{3b} = \omega^{-1}(\mathbf{b}_{3b} - \mathbf{z}_l^\mathsf{T}\mathbf{u}_3 - \mathbf{B}_{3b}^\mathsf{T}\mathbf{p}), \tag{C.4}$$

Next, we substitute (C.2)-(C.4) into the first to third and seventh equations in (C.1) and rearrange them. We get the local linear system reads as follows:

$$\begin{bmatrix} \mathbb{A} & \mathbb{B}_u^{\mathsf{T}} \\ \mathbb{B}_l & -\mathbb{E} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbb{b} \\ \mathbb{c} \end{bmatrix}, \tag{C.5}$$

where

$$\begin{aligned}
\mathbb{A} &= \begin{bmatrix} \mathbf{A} - \omega^{-1}\mathbf{z}_u\mathbf{z}_l^{\mathsf{T}} & 0 & 0 \\ 0 & \mathbf{A} - \omega^{-1}\mathbf{z}_u\mathbf{z}_l^{\mathsf{T}} & 0 \\ 0 & 0 & \mathbf{A} - \omega^{-1}\mathbf{z}_u\mathbf{z}_l^{\mathsf{T}} \end{bmatrix}, \\
\mathbb{B}_u &= \begin{bmatrix} \mathbf{B}_1 - \omega^{-1}\mathbf{z}_u\mathbf{B}_{1b}, & \mathbf{B}_2 - \omega^{-1}\mathbf{z}_u\mathbf{B}_{2b}, & \mathbf{B}_3 - \omega^{-1}\mathbf{z}_u\mathbf{B}_{3b} \end{bmatrix}, \\
\mathbb{B}_l &= \begin{bmatrix} \mathbf{B}_1 - \omega^{-1}\mathbf{z}_l^{\mathsf{T}}\mathbf{B}_{1b}, & \mathbf{B}_2 - \omega^{-1}\mathbf{z}_l^{\mathsf{T}}\mathbf{B}_{2b}, & \mathbf{B}_3 - \omega^{-1}\mathbf{z}_l^{\mathsf{T}}\mathbf{B}_{3b} \end{bmatrix}, \\
\mathbb{E} &= \omega^{-1}(\mathbf{B}_1\mathbf{B}_{1b}^{\mathsf{T}} + \mathbf{B}_2\mathbf{B}_{2b}^{\mathsf{T}} + \mathbf{B}_3\mathbf{B}_{3b}^{\mathsf{T}}), \\
\mathbb{b} &= \begin{bmatrix} \mathbf{b}_1 - \omega^{-1}\mathbf{z}_u\mathbf{b}_{1b} \\ \mathbf{b}_2 - \omega^{-1}\mathbf{z}_u\mathbf{b}_{2b} \\ \mathbf{b}_3 - \omega^{-1}\mathbf{z}_u\mathbf{b}_{3b} \end{bmatrix}, \\
\mathbb{c} &= -\omega^{-1}(\mathbf{b}_{1b}\mathbf{b}_{1b} + \mathbf{b}_{2b}\mathbf{b}_{2b} + \mathbf{b}_{3b}\mathbf{b}_{3b})
\end{aligned}$$

and $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)^{\mathsf{T}}$.

Assembly codes for three dimension

Non-vectorized and vectorized codes 3 and 4 follow. For non-vectorized code 3, no optimization was performed because it is unnecessary due to the high speed of the vectorized code 4.

```
function [A,Bl,Bu,E,b,c]=assembly3DP1bP1C(p,t,alpha,nu,f1,f2,f3,w)
% Assembly the linear system
%|A  Bu'||u|=|b|
%|Bl -E ||p|=|c|
% Matrices are permuted by variable t3 to format as follows:
% [ux_1,uy_1,uz_1,ux_2,uy_2,uz_2,...,ux_n,uy_n,uz_n,p_1,p_2,...,p_n]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   np=size(p,1); nt=size(t,1);
   Ah=sparse(np,np);  Z=sparse(np,np);
   Bu1=sparse(np,np); Bu2=sparse(np,np); Bu3=sparse(np,np);
   Bl1=sparse(np,np); Bl2=sparse(np,np); Bl3=sparse(np,np);
   E=sparse(np,np);
   b1=zeros(np,1); b2=zeros(np,1); b3=zeros(np,1); c=zeros(np,1);
   w1=w(1:3:3*np-2); w2=w(2:3:3*np-1); w3=w(3:3:3*np);
   Mel=(1/20)*[2 1 1 1; 1 2 1 1 ; 1 1 2 1; 1 1 1 2];
   for k=1:nt
      idt=t(k,:);
      % Tetrahedron volume
      x21=p(t(k,2),1)-p(t(k,1),1); y21=p(t(k,2),2)-p(t(k,1),2);
      x31=p(t(k,3),1)-p(t(k,1),1); y31=p(t(k,3),2)-p(t(k,1),2);
      x41=p(t(k,4),1)-p(t(k,1),1); y41=p(t(k,4),2)-p(t(k,1),2);
      x32=p(t(k,3),1)-p(t(k,2),1); y32=p(t(k,3),2)-p(t(k,2),2);
      x42=p(t(k,4),1)-p(t(k,2),1); y42=p(t(k,4),2)-p(t(k,2),2);
      z21=p(t(k,2),3)-p(t(k,1),3);
```

```
      z31=p(t(k,3),3)-p(t(k,1),3);
      z41=p(t(k,4),3)-p(t(k,1),3);
      z32=p(t(k,3),3)-p(t(k,2),3);
      z42=p(t(k,4),3)-p(t(k,2),3);
      xt=[y42*z32-y32*z42;y31*z41-y41*z31;y41*z21-y21*z41;y21*z31-y31*z21];
      yt=[x32*z42-x42*z32;z31*x41-z41*x31;z41*x21-z21*x41;z21*x31-z31*x21];
      zt=[x42*y32-x32*y42;x31*y41-x41*y31;x41*y21-x21*y41;x21*y31-x31*y21];
      tvolume=(x21*xt(2)+x31*xt(3)+x41*xt(4))/6;
      msgn=sign(tvolume); tvolume=abs(tvolume);
      % Sub calculations
      w1t=(w1(idt(1))+w1(idt(2))+w1(idt(3))+w1(idt(4)))/4;
      w2t=(w2(idt(1))+w2(idt(2))+w2(idt(3))+w2(idt(4)))/4;
      w3t=(w3(idt(1))+w3(idt(2))+w3(idt(3))+w3(idt(4)))/4;
      xyzt=xt(2)*xt(3)+yt(2)*yt(3)+zt(2)*zt(3)+xt(2)*xt(4)+yt(2)*yt(4)+zt(2)*zt(4)...
                      +xt(3)*xt(4)+yt(3)*yt(4)+zt(3)*zt(4);
      mt=(8/105)*alpha*tvolume*[1;1;1;1];
      omega_M=(621/3940)*alpha*tvolume;
      omega_R=(759/3152)*nu*(xt(1)^2+yt(1)^2+zt(1)^2-xyzt)/tvolume;
      omega=omega_M+omega_R;
      cut=(16/315)*(w1t*xt+w2t*yt+w3t*zt);
      comega=[1,2,2,2;
              2,1,2,2;
              2,2,1,2;
              2,2,2,1];
      zu=mt+comega*cut;
      zl=mt+(16/315)*(w1t*xt+w2t*yt+w3t*zt);
      f1t=(f1(idt(1))+f1(idt(2))+f1(idt(3))+f1(idt(4)))/4;
      f2t=(f2(idt(1))+f2(idt(2))+f2(idt(3))+f2(idt(4)))/4;
      f3t=(f3(idt(1))+f3(idt(2))+f3(idt(3))+f3(idt(4)))/4;
      f1b=(32/105)*f1t*tvolume;
      f2b=(32/105)*f2t*tvolume;
      f3b=(32/105)*f3t*tvolume;
      % Assembly matrices
      Ah(idt,idt)=Ah(idt,idt)+alpha*tvolume*Mel... % Mass matrix
                             +(nu/36/tvolume)*(xt*xt'+yt*yt'+zt*zt')... % Stiffness
                             +[1;1;1;1]*(w1t*xt'+w2t*yt'+w3t*zt')/24 ... % Convective
                             -zu*zl'/omega; % Bubble elimination
      Bu1(idt,idt)=Bu1(idt,idt)-[1;1;1;1]*msgn*xt'/24-(16/315)*msgn*(xt*zu')/omega;
      Bu2(idt,idt)=Bu2(idt,idt)-[1;1;1;1]*msgn*yt'/24-(16/315)*msgn*(yt*zu')/omega;
      Bu3(idt,idt)=Bu3(idt,idt)-[1;1;1;1]*msgn*zt'/24-(16/315)*msgn*(zt*zu')/omega;
      Bl1(idt,idt)=Bl1(idt,idt)-[1;1;1;1]*msgn*xt'/24-(16/315)*msgn*(xt*zl')/omega;
      Bl2(idt,idt)=Bl2(idt,idt)-[1;1;1;1]*msgn*yt'/24-(16/315)*msgn*(yt*zl')/omega;
      Bl3(idt,idt)=Bl3(idt,idt)-[1;1;1;1]*msgn*zt'/24-(16/315)*msgn*(zt*zl')/omega;
      E(idt,idt)=E(idt,idt)-(256/99225)*(xt*xt'+yt*yt'+zt*zt')/omega;
      % Assembly vectors
      b1(idt)=b1(idt)+f1t*tvolume*[1;1;1;1]/4-(zu*f1b)/omega;
      b2(idt)=b2(idt)+f2t*tvolume*[1;1;1;1]/4-(zu*f2b)/omega;
      b3(idt)=b3(idt)+f3t*tvolume*[1;1;1;1]/4-(zu*f3b)/omega;
      c(idt)=c(idt)-(16/315)*(xt*f1b+yt*f2b+zt*f3b)/omega;
   end
   t3(1:3:3*np-2)=1:np; t3(2:3:3*np-1)=np+[1:np]; t3(3:3:3*np)=2*np+[1:np];
   A=[Ah Z Z;
       Z Ah Z;
       Z Z Ah];
   A=A(t3,t3);
   Bu=[Bu1 Bu2 Bu3]; Bu=Bu(:,t3);
   Bl=[Bl1 Bl2 Bl3]; Bl=Bl(:,t3);
   b=[b1;b2;b3]; b=b(t3);
end
```

Listing 3: Non-vectorized 3D assembly function

```
function [A,Bl,Bu,E,b,c]=assembly3DP1bP1C_vec(p,t,alpha,nu,f1,f2,f3,w)
% Assembly the linear system
%|A  Bu'||u|=|b|
%|Bl -E ||p|=|c|
% Matrices are permuted by variable t3 to format as follows:
% [ux_1,uy_1,uz_1,ux_2,uy_2,uz_2,...,ux_n,uy_n,uz_n,p_1,p_2,...,p_n]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   np=size(p,1);
   Ah=sparse(np,np);  Z=sparse(np,np);
   Bu1=sparse(np,np); Bu2=sparse(np,np); Bu3=sparse(np,np);
   Bl1=sparse(np,np); Bl2=sparse(np,np); Bl3=sparse(np,np);
   E=sparse(np,np);
   b1=sparse(np,1); b2=sparse(np,1); b3=sparse(np,1); c=sparse(np,1);
   % Tetrahedrons volume
   x21=p(t(:,2),1)-p(t(:,1),1); y21=p(t(:,2),2)-p(t(:,1),2);
   x31=p(t(:,3),1)-p(t(:,1),1); y31=p(t(:,3),2)-p(t(:,1),2);
   x41=p(t(:,4),1)-p(t(:,1),1); y41=p(t(:,4),2)-p(t(:,1),2);
   x32=p(t(:,3),1)-p(t(:,2),1); y32=p(t(:,3),2)-p(t(:,2),2);
   x42=p(t(:,4),1)-p(t(:,2),1); y42=p(t(:,4),2)-p(t(:,2),2);
   z21=p(t(:,2),3)-p(t(:,1),3);
   z31=p(t(:,3),3)-p(t(:,1),3);
   z41=p(t(:,4),3)-p(t(:,1),3);
   z32=p(t(:,3),3)-p(t(:,2),3);
   z42=p(t(:,4),3)-p(t(:,2),3);
   xt=[z32.*y42-y32.*z42,y31.*z41-z31.*y41,z21.*y41-y21.*z41,y21.*z31-z21.*y31];
   yt=[x32.*z42-z32.*x42,z31.*x41-x31.*z41,x21.*z41-z21.*x41,z21.*x31-x21.*z31];
   zt=[y32.*x42-x32.*y42,x31.*y41-y31.*x41,y21.*x41-x21.*y41,x21.*y31-y21.*x31];
   tvolume=(x21.*xt(:,2)+x31.*xt(:,3)+x41.*xt(:,4))/6;
   msgn=sign(tvolume); tvolume=abs(tvolume);
   % Sub calculations
   w1=w(1:3:3*np-2); w2=w(2:3:3*np-1); w3=w(3:3:3*np);
   w1t=(w1(t(:,1))+w1(t(:,2))+w1(t(:,3))+w1(t(:,4)))/4;
   w2t=(w2(t(:,1))+w2(t(:,2))+w2(t(:,3))+w2(t(:,4)))/4;
   w3t=(w3(t(:,1))+w3(t(:,2))+w3(t(:,3))+w3(t(:,4)))/4;
   w1xw2yw3z=[w1t.*xt(:,1)+w2t.*yt(:,1)+w3t.*zt(:,1),...
              w1t.*xt(:,2)+w2t.*yt(:,2)+w3t.*zt(:,2),...
              w1t.*xt(:,3)+w2t.*yt(:,3)+w3t.*zt(:,3),...
              w1t.*xt(:,4)+w2t.*yt(:,4)+w3t.*zt(:,4)];
   xtx=xt(:,2).*xt(:,3)+xt(:,2).*xt(:,4)+xt(:,3).*xt(:,4);
   xty=yt(:,2).*yt(:,3)+yt(:,2).*yt(:,4)+yt(:,3).*yt(:,4);
   xtz=zt(:,2).*zt(:,3)+zt(:,2).*zt(:,4)+zt(:,3).*zt(:,4);
   xtt=xtx+xty+xtz;
   mt=(8/105)*alpha*tvolume;
   omega_M=(621/3940)*alpha*tvolume;
   omega_R=(759/3152)*nu*(xt(:,1).^2+yt(:,1).^2+zt(:,1).^2-xtt)./tvolume;
   omega=omega_M+omega_R;
   zu(:,1)=mt+(16/315)*(w1xw2yw3z(:,1)+2*w1xw2yw3z(:,2)...
                                      +2*w1xw2yw3z(:,3)+2*w1xw2yw3z(:,4));
   zu(:,2)=mt+(16/315)*(2*w1xw2yw3z(:,1)+w1xw2yw3z(:,2)...
                                        +2*w1xw2yw3z(:,3)+2*w1xw2yw3z(:,4));
   zu(:,3)=mt+(16/315)*(2*w1xw2yw3z(:,1)+2*w1xw2yw3z(:,2)...
                                        +w1xw2yw3z(:,3)+2*w1xw2yw3z(:,4));
   zu(:,4)=mt+(16/315)*(2*w1xw2yw3z(:,1)+2*w1xw2yw3z(:,2)...
                                        +2*w1xw2yw3z(:,3)+w1xw2yw3z(:,4));
   zl=mt+(16/315)*(w1t.*xt+w2t.*yt+w3t.*zt);

   f1t=(f1(t(:,1))+f1(t(:,2))+f1(t(:,3))+f1(t(:,4)))/4;
   f2t=(f2(t(:,1))+f2(t(:,2))+f2(t(:,3))+f2(t(:,4)))/4;
   f3t=(f3(t(:,1))+f3(t(:,2))+f3(t(:,3))+f3(t(:,4)))/4;
   f1b=(32/105)*tvolume.*f1t;
   f2b=(32/105)*tvolume.*f2t;
   f3b=(32/105)*tvolume.*f3t;
```

```
   % Mass matrix
   for i=1:4
      for j=1:i
         Ah=Ah+sparse(t(:,i),t(:,j),alpha*tvolume/20,np,np);
      end
   end
   Ah=Ah+Ah.';
   % Stiffness matrix
   for i=1:4
      for j=1:4
         Ah=Ah+sparse(t(:,i),t(:,j),(nu/36)*(xt(:,i).*xt(:,j)+yt(:,i).*yt(:,j)...
                                             +zt(:,i).*zt(:,j))./tvolume,np,np);
      end
   end
   % Main part of the convective matrix
   cal=(1/24)*w1xw2yw3z;
   for i=1:4
      for j=1:4
         Ah=Ah+sparse(t(:,i),t(:,j),cal(:,j),np,np);
      end
   end
   for i=1:4
      % Assembly vectors
      b1=b1+sparse(t(:,i),1,(1/4)*f1t.*tvolume-zu(:,i).*f1b./omega,np,1);
      b2=b2+sparse(t(:,i),1,(1/4)*f2t.*tvolume-zu(:,i).*f2b./omega,np,1);
      b3=b3+sparse(t(:,i),1,(1/4)*f3t.*tvolume-zu(:,i).*f3b./omega,np,1);
      c=c-sparse(t(:,i),1,(16/315)*(xt(:,i).*f1b+yt(:,i).*f2b ...
                                              +zt(:,i).*f3b)./omega,np,1);
      for j=1:4
         % Assembly matrices
         Ah=Ah-sparse(t(:,i),t(:,j),zu(:,i).*zl(:,j)./omega,np,np);
         Bu1=Bu1-sparse(t(:,i),t(:,j),(1/24)*msgn.*xt(:,j)...
                                      +(16/315)*msgn.*zu(:,j).*xt(:,i)./omega,np,np);
         Bu2=Bu2-sparse(t(:,i),t(:,j),(1/24)*msgn.*yt(:,j)...
                                      +(16/315)*msgn.*zu(:,j).*yt(:,i)./omega,np,np);
         Bu3=Bu3-sparse(t(:,i),t(:,j),(1/24)*msgn.*zt(:,j)...
                                      +(16/315)*msgn.*zu(:,j).*zt(:,i)./omega,np,np);
         Bl1=Bl1-sparse(t(:,i),t(:,j),(1/24)*msgn.*xt(:,j)...
                                      +(16/315)*msgn.*zl(:,j).*xt(:,i)./omega,np,np);
         Bl2=Bl2-sparse(t(:,i),t(:,j),(1/24)*msgn.*yt(:,j)...
                                      +(16/315)*msgn.*zl(:,j).*yt(:,i)./omega,np,np);
         Bl3=Bl3-sparse(t(:,i),t(:,j),(1/24)*msgn.*zt(:,j)...
                                      +(16/315)*msgn.*zl(:,j).*zt(:,i)./omega,np,np);
         E=E-sparse(t(:,i),t(:,j),(256/99225)*(xt(:,i).*xt(:,j)+yt(:,i).*yt(:,j)...
                                            +zt(:,i).*zt(:,j))./omega,np,np);
      end
   end
   t3(1:3:3*np-2)=1:np; t3(2:3:3*np-1)=np+[1:np]; t3(3:3:3*np)=2*np+[1:np];
   A=[Ah Z Z;
       Z Ah Z;
       Z  Z Ah];
   A=A(t3,t3);
   Bu=[Bu1 Bu2 Bu3]; Bu=Bu(:,t3);
   Bl=[Bl1 Bl2 Bl3]; Bl=Bl(:,t3);
   b=[full(b1); full(b2); full(b3)]; b=b(t3);
   c=full(c);
end
```

Listing 4: Vectorized 3D assembly function